CREATING A GENERIC HAND AND FINGER GESTURE RECOGNIZER BY USING FOREARM MUSCLE ACTIVITY SIGNALS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

UMUT DEMİREL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MULTIMEDIA INFORMATICS

AUGUST 2017

Approval of the thesis:

CREATING A GENERIC HAND AND FINGER GESTURE RECOGNIZER BY USING FOREARM MUSCLE ACTIVITY SIGNALS

submitted by **UMUT DEMİREL** in partial fulfillment of the requirements for the degree of **Master of Science in Modelling and Simulation Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Director, **Graduate School of Informatics, METU**

Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu
Head of Department, **Modelling and Simulation, METU**

Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu
Supervisor, **Modelling and Simulation, METU**

Assist. Prof. Dr. Elif Sürer
Co-supervisor, **Modelling and Simulation, METU**

Examining Committee Members:

Prof. Dr. Yasemin Yardımcı Çetin
Information Systems, METU

Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu
Modelling and Simulation, METU

Assist. Prof. Dr. Elif Sürer
Modelling and Simulation, METU

Assoc. Prof. Dr. Alptekin Temizel
Modelling and Simulation, METU

Assist. Prof. Dr. Cihan Topal
Electrical and Electronics Engineering, Anadolu University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: UMUT DEMİREL

Signature :

ABSTRACT

CREATING A GENERIC HAND AND FINGER GESTURE RECOGNIZER
BY USING FOREARM MUSCLE ACTIVITY SIGNALS

DEMİREL, UMUT
M.S., Department of Modelling and Simulation
Supervisor : Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu
Co-Supervisor : Assist. Prof. Dr. Elif Sürer

August 2017, 87 pages

Hand and finger gestures are one of the most natural ways of non-verbal communication. Apart from their daily use in different cultures, they are also widely used in human-computer interaction. There are a variety of applications using gestures as inputs such as sign language recognition, robot control, mobile phone control, medical device control and video game control. Advances in near-field wireless communications made it possible to design and deploy low-cost, inconspicuous control devices which can be used to detect certain predefined hand gestures for use in interaction. This thesis aims to investigate and develop a generic hand and finger gesture recognizer by processing forearm muscle activity signals from such a device which consists of eight electromyography (EMG) and Inertial Measurement Unit (IMU) sensors. Two main presuppositions of this thesis is that i) the muscle activity on the forearm is a spatiotemporally bandlimited circular signal and that can be sampled using a finite number of sensors, and ii) different gestures result in different but consistent patterns which are separable. The approach used in this thesis is based on the extraction of features by joint processing of signals obtained from a commercially available, low-cost EMG armband and classification of the gestures by simple and low-complexity artificial neural networks (ANNs). The dictionary of gestures were chosen from a canonical catalog of expressive gestures of classical orchestra conductors. Two experiments were carried out to assess the system performance: i) thirteen different hand and finger gestures and one rest gesture are performed 5 times in the same session by 10 different subjects, and ii) data was collected in an ecological study from a conductor during a practice session of a symphony orchestra. It was found that the proposed method achieved an average classification accuracy of 63.14% (maximum of 79.87%) when the data distribution for train, test and validate parts of ANN used in classification process is separated by sessions. An average classification accuracy of 96.09% (maximum of 98.8%) was achieved when data distribution is random. Lastly, random data distribution of the five different gestures and one rest gesture data collected in an ecological study from a conductor resulted in 96.9% accuracy. All results were obtained with session and subject dependent experiments.

Keywords: gesture recognition, electromyogram, muscle activity, neural networks, circular harmonic transform

ÖZ

ÖN KOL KAS HAREKETLERİNDEN OLUŞAN SİNYALLERİ KULLANARAK EL VE PARMAK İŞARETLERİNİ TANIYAN JENERİK BİR SİSTEM GELİŞTİRME

DEMİREL, UMUT
Yüksek Lisans, Oyun Teknolojileri Bölümü
Tez Yöneticisi : Doç. Dr. Hüseyin Hacıhabiboğlu
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Elif Sürer

Ağustos 2017 , 87 sayfa

El ve parmak işaretleri, konuşmadan iletişim kurmanın en doğal yollarından biridir. Bu işaretler farklı kültürlerde günlük yaşamda iletişim kurmak için kullanılmalarının yanısıra, bilişim dünyasında insan-bilgisayar etkileşimi alanında da sıklıkla kullanılmaktadır. El ve parmak işaret verileri; işaret dili tanıma, robot kontrol etme, cep telefonu kontrol etme, medikal araç kontrol etme ve video oyunu kontrol etme gibi alanlarda geliştirilen uygulamalarda girdi olarak kullanılmaktadır. Yakın-alan kablosuz iletişim alanındaki gelişmeler, önceden tanımlanmış el ve parmak işaretlerini tanıyan ucuz maliyetli ve küçük boyutlu kontrol cihazlarını tasarlamayı olanaklı kılmaktadır. Bu tez kapsamında, ön kol kas aktivitelerini ve hareketlerini ölçmek için 8 elektromiyografi (EMG) sensörü ve eylemsizlik ölçüm unitesi (IMU) kullanılarak genel bir işaret tanıma sistemi geliştirilmesi amaçlanmıştır. Bu tezin iki önvarsayımı i) ön kol kas aktiviteleri zaman-uzamsal limitli dairesel sinyallerdir ve bu sinyaller sonlu sayıdaki sensörler ile örneklendirilebilir, ve ii) farklı el ve parmak işaretleri, ayırt edilebilmeyi sağlayan birbirinden farklı ancak tutarlı modellerle temsil edilebilir. Bu tezde kullanılan yaklaşım, ticari olarak erişilebilir düşük maliyetli EMG kol bandı kullanılarak elde edilen sinyallerin işlenmesiyle belirli özelliklerin özütlenmesini ve basit yapay sinirsel ağlar kullanarak el ve parmak işaretlerinin sınıflandırılmasını temel almaktadır. El ve parmak işaretleri kümesi, klasik müzik orkestra şeflerinin dışavurumcu olarak kullandıkları el işaretlerinden oluşturulmuştur. Sistem performansını ölçmek amacıyla iki deney çalışması düzenlenmiştir: i) 13 farklı el ve parmak işareti ve bir dinlenme pozisyonu aynı seans içinde 5er defa 10 test kullanıcısı tarafından gerçekleştirilerek veri toplanmıştır, ve ii) bir klasik müzik konser provasında orkestra şefi gerçek hareketlerini gerçekleştirirken veri toplanmıştır. Kullanılan yapay sinir ağını eğitirken, onaylarken ve test ederken ağın beslenmesi için kullanılan verinin dağılımının seanslara göre yapılması sonucu ortalama %63.14 (en yüksek %79.87) başarı oranı yakaladığı saptanmıştır. Veri dağılımının rastgele yapılması sonucunda ise ortalama %96.09 (en yüksek %98.8) başarı oranı yakalandığı görülmüştür. Son olarak, orkestra şefinden toplanan 5 farklı el ve parmak işareti ve bir dinlenme pozisyonu verilerinin yapay sinir ağlarına rastgele dağılımı sonucunda %96.9 başarı oranı yakalanmıştır. Bütün sonuçlara, kişiye ve seansa bağlı olan deneyler düzenlenerek erişilmiştir.

Anahtar Kelimeler: işaret tanıma, elektromiyogram, kas aktivitesi, sinirsel ağlar, çembersel harmonik dönüşüm

Dedicated to the %50 of this beautiful country...

ACKNOWLEDGMENTS

First and foremost I would like to express my gratefulness to my precious thesis advisor Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu. His patience, understanding and supervision gave a great shape to this thesis which eventually led to success. He has been encouraging me to complete this work for more than 3 years. He always made me feel that he believed in me. Thank you, seriously.

Secondly, I would like to thank to my dear thesis co-supervisor Assist. Prof. Dr. Elif Sürer. She also has been motivating me during the whole thesis period. Her valuable comments had a great impact on improving this study.

I also would like to thank to Prof. Dr. Yasemin Yardımcı Çetin, Assoc. Prof. Dr. Alptekin Temizel and Assist. Prof. Dr. Cihan Topal for accepting to participate in the precious jury members of my thesis.

Furthermore, my appreciation goes to Assoc. Prof. Dr. Fazlı Orhun Orhon from Başkent University State Conservatory. He allowed us to collect gesture data from himself by wearing Myo device during an important concert practice, which enabled us to create a great data set for the experiments of this study.

In addition to them, I would like to thank to my former business partner Kutay Yalçınkaya who took control of our company during my thesis work. He never refused me to put Myo on his forearm when I said that I needed to collect test data.

Of course my dear family… My father and mother always motivated me to get my master's degree. They always made me feel that they were proud of me, my successes and my fails. I wish they asked when I would finish writing my thesis and graduate less time but anyway, I have been sending huge thanks from deep in my heart and I want to say that I love them so much.

Lastly, I would like to thank to my wife, Eda Demirdüzen Demirel, the rest of my life. We met her before starting to work on this thesis, fell in love during the research and got married just before I completed my thesis work. Her support and academic advices for all parts of this study are inevitable. Thank you Eda, for being a part of me.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xiii

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xvii

CHAPTERS

1 INTRODUCTION 1

1.1 Body Language, Gestures and Postures 1

1.2 Human Computer Interaction (HCI) 2

1.3 Gesture Recognition 3

1.4 Forearm Muscles 3

1.5 Thesis Structure 5

2 BACKGROUND 9

2.1 Data Collection 10

2.2 Pre-processing 11

2.3 Feature Extraction 12

2.3.1 Raw or Pre-processed Data 12

2.3.2 Inter-Element Difference (IED) 13

2.3.3 Detected Markers 13

2.3.4 Mean Absolute Value (MAV) 13

2.3.5 Moving Averaging Window (MAW) 14

2.3.6 Root Mean Square (RMS) 14
2.3.7 Standard Deviation (STD) 14
2.3.8 Circular Harmonic Coefficients 15
2.4 Feature Reduction 16
2.4.1 Principal Component Analyses (PCA) 16
2.4.2 Singular Value Decomposition (SVD) 16
2.4.3 Random Projection (RP) 17
2.5 Classification 17
2.5.1 Artificial Neural Networks (ANN) 18
2.5.2 Hidden Markov Models (HMM) 19
2.5.3 Support Vector Machines (SVM) 21
2.5.4 K-Nearest Neighbor (K-NN) 22
3 PREVIOUS WORK 25
3.1 Vision Based Gesture Recognition Techniques 25
3.1.1 Camera and Color Glove 26
3.1.2 Camera and Bare Hand 28
3.1.3 Depth Camera 29
3.2 Sensor Based Gesture Recognition Techniques 30
3.2.1 Accelerometer Sensors 30
3.2.2 Flex Sensors 32
3.2.3 EMG Sensors 34
3.2.3.1 Only EMG Sensors 34
3.2.3.2 EMG + Accelerometer Sensors . . . 36
3.2.3.3 EMG + IMU Sensors 38
3.2.3.4 Myo 38
4 PROPOSED METHOD 43
4.1 Selecting Gesture Set 43
4.2 Collecting Data Using Myo 44
4.3 Pre-processing 47
4.4 Extracting Feature Vectors 48

4.4.1 Moving Averaging Window 49

4.4.2 FWR of Inter-Element Differences 49

4.4.3 Circular Harmonic Coefficients 51

4.5 Reducing the Number of Feature Vectors 52

4.6 Performing Classification 53

5 EXPERIMENTS AND RESULTS 55

5.1 System Validation 55

5.1.1 Using *dividerand* for ANN's data dividing function 56

5.1.2 Using *divideblock* for ANN's data dividing function 59

5.2 Basketball Referee Gestures 60

5.3 Classical Music Orchestra Conductor Gestures 62

6 CONCLUSION AND FUTURE WORK 65

6.1 Discussions 65

6.2 Contributions 71

6.3 Future Work 72

REFERENCES 73

APPENDICES

A CIRCULAR HARMONIC FUNCTION REPRESENTATIONS OF 14 GESTURES 83

B APPROVAL OF METU HUMAN SUBJECTS ETHICS COMMITTEE 87

LIST OF TABLES

Table 1.1 Actions of forearm muscles 4

Table 2.1 Types of data for each data collection method 10
Table 2.2 Transition probabilities for the given example 20
Table 2.3 Emission probabilities for the given example 20
Table 2.4 Euclidean distances between classified data points and the new point 23

Table 3.1 Active EMG sensors of Myo for each gesture 40

Table 5.1 Gesture names and corresponding abbreviations 55
Table 5.2 Classification results (in percentages) with 7 different training functions (PCA was applied and the first 14 PCs were used) 57
Table 5.3 Training time results (in seconds) with 7 different training functions (PCA was applied and the first 14 PCs were used) 57
Table 5.4 Classification results (in percentages) with 7 different training functions (PCA was applied and the first 40 PCs were used) 58
Table 5.5 Training time results (in seconds) with 7 different training functions (PCA was applied and the first 40 PCs were used) 58
Table 5.6 Classification and training time comparison of using 14 and 40 PCs 59
Table 5.7 Data set setups for ANN with *divideblock* dividing function . 60
Table 5.8 Minimum, maximum, and average results for cross testing . . 60
Table 5.9 Classification results of the proposed algorithm which was input with the basketball referee gesture data 61
Table 5.10 Classification results of LOPOCV 62
Table 5.11 Classification results of the proposed algorithm which was input with the conductor training data 64

Table 6.1 EMG channel setup of Myo rotation experiment 66
Table 6.2 Classification results of the proposed algorithm which was input with the rotation experiment data 66
Table 6.3 Classification results of the real-time rotation detection algorithm 67
Table 6.4 Comparison of the minimum, maximum, and average results for cross testing of 13+1 gesture data set and 5+1 gesture data set . . 68
Table 6.5 Mean values of each EMG channel shown in Figure 6.1 69

Table 6.6 Classification results with 1-second-gesture performances instead of 5-second-gesture performances 70
Table 6.7 Classification results with different feature sets 71

LIST OF FIGURES

Figure 1.1 Commercial HCI products 2

Figure 1.2 Superficial forearm muscles 6

Figure 1.3 Representation of Myo on superficial forearm muscles. Adopted from "Cross section of the muscles of the distal forearm. Some extensor muscles, such as the anconeus, are not visible as they are situated proximally in the forearm." by TeachMeAnatomy.com, retrieved from http://www.bartleby.com/107/Images/large/image417 .gif Used under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license (https://creativecommons.org/ licenses/by-nc-nd/4.0/) 7

Figure 2.1 Gesture recognition process 9

Figure 2.2 Artificial Neural Network Representation 18

Figure 2.3 HMM Example 19

Figure 2.4 All the calculations of Viterbi algorithm for given example . 20

Figure 2.5 Support Vector Machine Representation 21

Figure 2.6 Kernel Transformation Function Representation 22

Figure 2.7 K-NN Example 22

Figure 3.1 CG Examples 26

Figure 3.2 Accelerometer Device Examples 31

Figure 3.3 Glove Examples Having Flex Sensors 32

Figure 3.4 EMG reading differences of the same gesture by applying different levels of strength while performing gestures 40

Figure 4.1 Expressive gesture set of classical music orchestra conductor 44

Figure 4.2 Flow Diagram that explains the steps of data collection process 45

Figure 4.3 8-channel sample EMG signals for fist gesture 46

Figure 4.4 Signal plots after applying FWR to the data in Figure 4.3 . . 47

Figure 4.5 Signal plots after applying MAW to the data in Figure 4.4 . . 47

Figure 4.6 Signal plots after applying IED to the data in Figure 4.3 . . . 49

Figure 4.7 Signal plots after applying FWR to the data in Figure 4.6 . . 50

Figure 4.8 Signal plots after applying MAW to the data in Figure 4.7 . . 50

Figure 4.9 Myo representation on polar coordinate system 51

Figure 4.10 Comparison of the classification results with and without applying PCA 53

Figure 4.11 Resulting ANN with 14 principle components, 20 hidden layers and 14 outputs 54

Figure 5.1 A snapshot from the concert training captured with 3 different cameras 63

Figure 6.1 Signal plots of pre-processed fist gesture data of TS8 during 5 consecutive data collection sessions 69

LIST OF ABBREVIATIONS

1D	1 dimensional
2D	2 dimensional
3D	3 dimensional
ANN	Artificial Neural Networks
AR	Augmented Reality
ASL	American Sign Language
AVF	Absolute Value Filter
Avg	Average
CCW	Counter Clockwise
CG	Color Glove
CW	Clockwise
DBN	Deep Belief Network
DTW	Dynamic Time Warping
EMG	Electromyography
FFT	Fast Fourier Transform
fps	Frame per second
FWR	Full-Wave Rectification
HCI	Human Computer Interaction
HDD	Hard Disk Drive
HMM	Hidden Markov Model
HSI	Hue, Saturation, Intensity
IED	Inter-Element Differences
IMU	Inertial Measurement Unit
K-NN	K-Nearest Neighbor
kHz	Kilo Hertz
LED	Light-Emitting Diode
LOPOCV	Leave-One-Participant-Out-Cross-Validation
LVQ	Learning Vector Quantization
MAV	Mean Absolute Value
MAW	Moving Averaging Window
Max	Maximum
Min	Minimum

ms	milisecond
PCA	Principal Component Analysis
PCs	Principal Components
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machines
RC	Radio Controlled
RGB	Red, Green, Blue
RMS	Root Mean Square
RP	Random Projection
SSI	Simple Square Integral
STD	Standard Deviation
SUS	System Usability Scale
SVD	Singular Value Decomposition
SVM	Support Vector Machine
VCR	Videocassette Recorder
VR	Virtual Reality
wrt	with respect to

CHAPTER 1

INTRODUCTION

Communication has an essential role in the emergence and nourishment of social life. It allows humans to coordinate their behavior with that of others. The most obvious way of communication between individuals is spoken language [1].

Besides, body language can be used as a common tongue between humans to communicate without speaking. According to the results of the experiments, Ekman [2] found out that although most of the facial expressions of emotions are culture-specific, there is also a universal set of facial expressions of emotion.

1.1 Body Language, Gestures and Postures

Body Language, also known as Kinesics is defined by Birdwhistell as

> "Systematic study of visually sensible aspects of non-verbal interpersonal communication"

in 1955 [3].

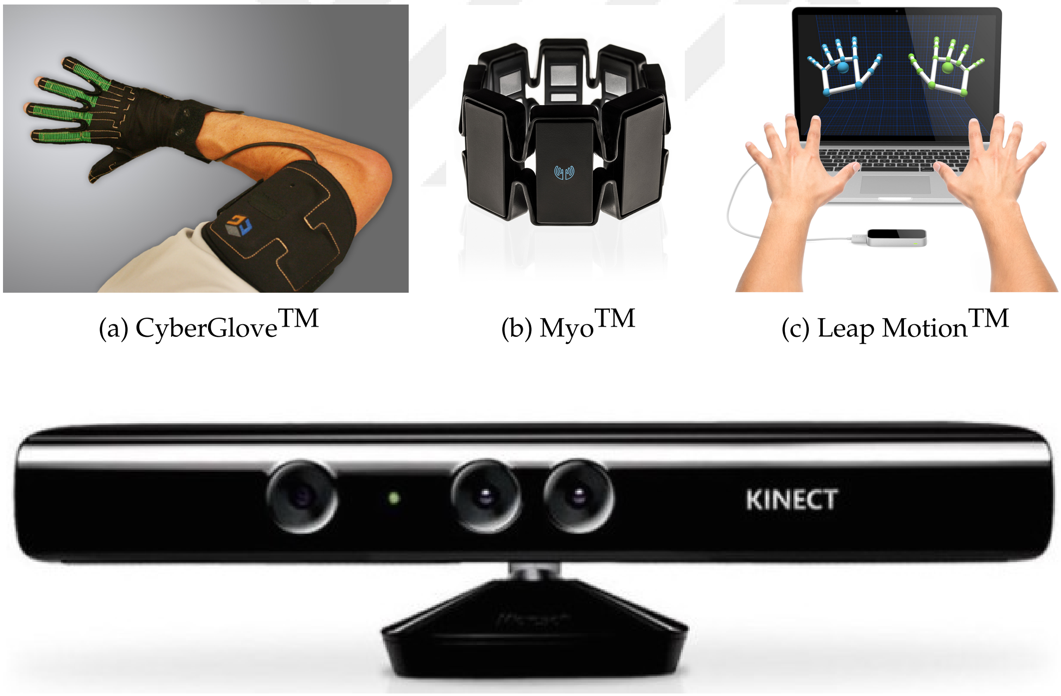
Another definition of body language came from Soukhanov in 1992 as

> “[Body language consists of] the bodily gestures, postures, and facial expressions by which a person communicates nonverbally with others.”

Examples of bodily gestures are shrugging, raising arms or rotating head. Instances of postures can be bending the whole body or angular distance of the upper body. Facial expression representations can be raising eyebrow, blinking eye or smiling. Soukhanov goes on to define "gesture" as a sign, signal, or cue used to communicate in tandem with, or apart from words [4]. Gestures can be used not only for expressing feelings but also for giving messages to others. For example, raising a hand with the palms towards someone means “stop” or shaking the hand with fingers spread means “Goodbye” in many cultures. For this example, "stop" sign is a posture of hand and it becomes a conceptual gesture because it is used for communication. "goodbye" gesture is formed with the combination of a hand posture and a hand movement.

1.2 Human Computer Interaction (HCI)

Unprecedented advances in information and communication technologies and the need for more natural interaction modalities brought gesture and posture at the forefront as a possible means for controlling computing devices. Gesture-controlled interfaces that were the stock of science fiction works as seen in movies like "Minority Report" [5] have been achieved earlier than predicted. Commercially available products such as Microsoft Kinect [6], CyberGlove [7], Leap Motion [8] or Myo [9] were placed in the market together with their own applications as well as SDKs for wider adoption. With their affordable prices, these products can be used in end-user entertainment applications like games, Augmented Reality (AR) and Virtual Reality (VR) applications or controlling robots. Besides, these devices provide opportunities for researchers to collect data for gesture recognition.



(a) CyberGlove^{TM}

(b) Myo^{TM}

(c) Leap Motion^{TM}

(d) Microsoft Kinect^{TM}

Figure 1.1: Commercial HCI products

While many different signal processing and computer vision algorithms have been developed to detect different types of gestures, technology companies have been manufacturing cameras and portable devices making data collection process much easier. Thanks to the portability and integrability of Arduino platform [10] and its derivatives such as BITalino [11], researchers are now able to create their own device prototypes by combining different types of sensors [12, 13, 14, 15].

1.3 Gesture Recognition

Human hand can be moved to generate many different shapes and signs and it is one of the most effective organ as a general purpose interaction tool [16, 17]. Moreover, using hand and finger gestures as inputs is widely used in human-computer interaction applications. There are variety of applications using gestures as inputs such as sign language recognition, hand tracking, robot control, mobile phone control, medical device control, and video game control. The reason why gestures may be more preferable than other options is that they can be robustly recognised by using different techniques providing application-level flexibility. These techniques can be categorized into two groups: vision based gesture recognition techniques and sensor based gesture recognition techniques. For vision based gesture recognition techniques, shape of the hand and fingers can be detected from a video stream. One such approach involves the segmentation of the hand to its constituent parts such as the palm and the fingers by using colored gloves [18, 19, 20, 21, 22, 23]. Detection can also be done with regular, infrared or depth cameras and which both require image/video processing techniques. For sensor based gesture recognition techniques; flex sensors (that change their impedance in response to flexure) for finger movements; inertial measurement unit (accelerometer, gyroscope and magnetometer) for hand movements, rotations and directions; electromyography (EMG) sensors for muscle activity detection can be used. Data collection can be performed either with sensors placed on fingers and arms or with commercialized data gloves and armbands which accommodate the combination of some of the sensors mentioned above.

The sensor based approach typically involves measurements over the forearm. Since the approach proposed in this thesis is also based on EMG signals acquired over the forearm, information on the forearm muscle structure would be helpful. This is covered in the next section.

1.4 Forearm Muscles

Different gestures activate different muscles on the forearm. Posterior and anterior superficial forearm muscles [24, 25] can be seen in Figure 1.2. Table 1.1 explains responsibilities of each superficial forearm muscle.

To be more clear, anatomical definitions of muscle actions [26] are given as follows:

- **Flexion** describes a bending movement that decreases the angle between

Table 1.1: Actions of forearm muscles

Muscle Name		Action
Posterior	Brachioradialis	Flexes at the elbow
	Extensor carpi radialis longus and brevis	Extends and abducts the wrist
	Extensor digitorum	Extends medial four fingers at the MCP and IP joints
	Extensor digit minimi	Extends the little finger and contributes to extension at the wrist
	Extensor carpi ulnaris	Extension and adduction of wrist
	Anconeus	Moves the ulna during pronation and extends at the elbow joint
Anterior	Flexor carpi ulnaris	Flexion and adduction at the wrist
	Palmaris longus	Flexion at the wrist
	Flexor carpi radialis	Flexion and abduction at the wrist
	Pronator teres	Pronation of the forearm

a segment and its proximal segment [27].

- **Extension** is the opposite of flexion, describing a straightening movement that increases the angle between body parts [27].

- **Abduction** refers to a motion that pulls a structure or part away from the midline of the body. Abduction of the wrist is also called radial deviation [27].

- **Adduction** refers to a motion that pulls a structure or part toward the midline of the body, or towards the midline of a limb. Adduction of the wrist is also called ulnar deviation [27].

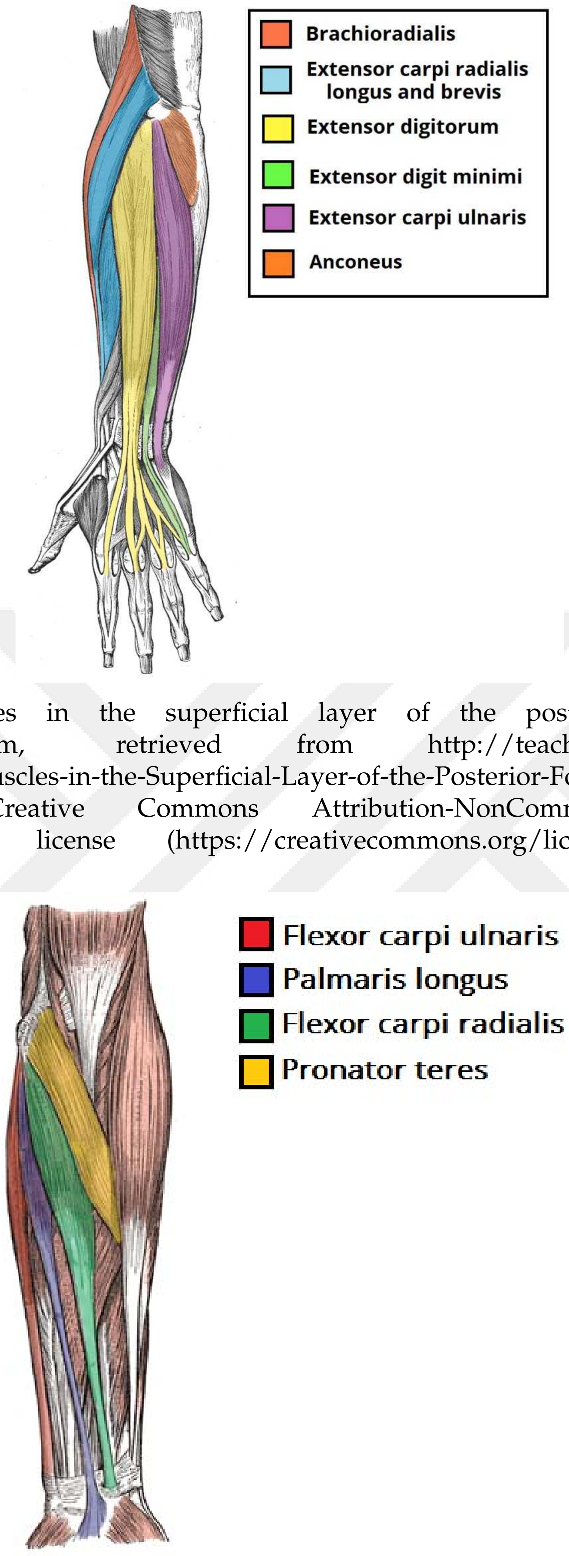
- **Pronation** at the forearm is a rotational movement where the hand and upper arm are turned inwards [27].

Having the information of which muscles are activated while performing different gestures is important when EMG data is used for gesture recognition. By placing electrodes of EMG sensors on specific muscles, gestures activating different muscles can be detected. Good positions for electrodes can be specified by trying to place electrodes on different parts of forearm and evaluating the level of the obtained signal (for example when a certain gesture is performed). This approach is possible when using EMG devices with separate electrodes. However, if the EMG device is an armband with EMG sensors located in a circular order, accurate positioning on forearm muscles may not be possible. The EMG armband, Myo, used in this thesis is an example of such a sensor array. A representation of Myo on superficial forearm muscles can be seen in Figure 1.3). Even small rotations of the armband on the forearm may cause different signal patterns.

1.5 Thesis Structure

The work described in this thesis involves the development of a generic hand and finger gesture recognizer based on the joint-processing of EMG signals which represent muscle activity. For data collection, Myo from Thalmic Labs, a portable armband having 8 circularly distributed EMG sensors and an IMU, is used. Placement positions of the EMG sensors on the forearm is important for data acquisition under ideal conditions in the lab. However, the purpose of this thesis is to make it possible to acquire signals and classify gestures under less than ideal, ecological conditions (such as a user using the system at home). Therefore, the proposed system requires calibration and is person and session dependent.

The thesis consists of six chapters including this chapter. Gesture types and steps of gesture recognition process are explained in the "Chapter 2: Background". These are followed by the discussion of the gesture recognition usage areas and gesture recognition techniques together with the great examples in "Chapter 3: Previous Work". In "Chapter 4: Proposed Method", the proposed solution is explained and studied. Conducted experiments and corresponding results of the study are discussed in "Chapter 5: Experiments and Results". Finally in "Chapter 6: Conclusion and Future Work", shortcomings of the proposed system are discussed, the conclusions are made and possible future directions are pointed.



(a) "The muscles in the superficial layer of the posterior forearm." by TeachMeAnatomy.com, retrieved from http://teachmeanatomy.info/wp-content/uploads/Muscles-in-the-Superficial-Layer-of-the-Posterior-Forearm-824x1024.jpg Used under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

(b) "The superficial muscles of the anterior forearm." by TeachMeAnatomy.com, retrieved from http://teachmeanatomy.info/wp-content/uploads/Superficial-Flexor-Muscles-of-the-Anterior-Forearm.jpg Used under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Figure 1.2: Superficial forearm muscles

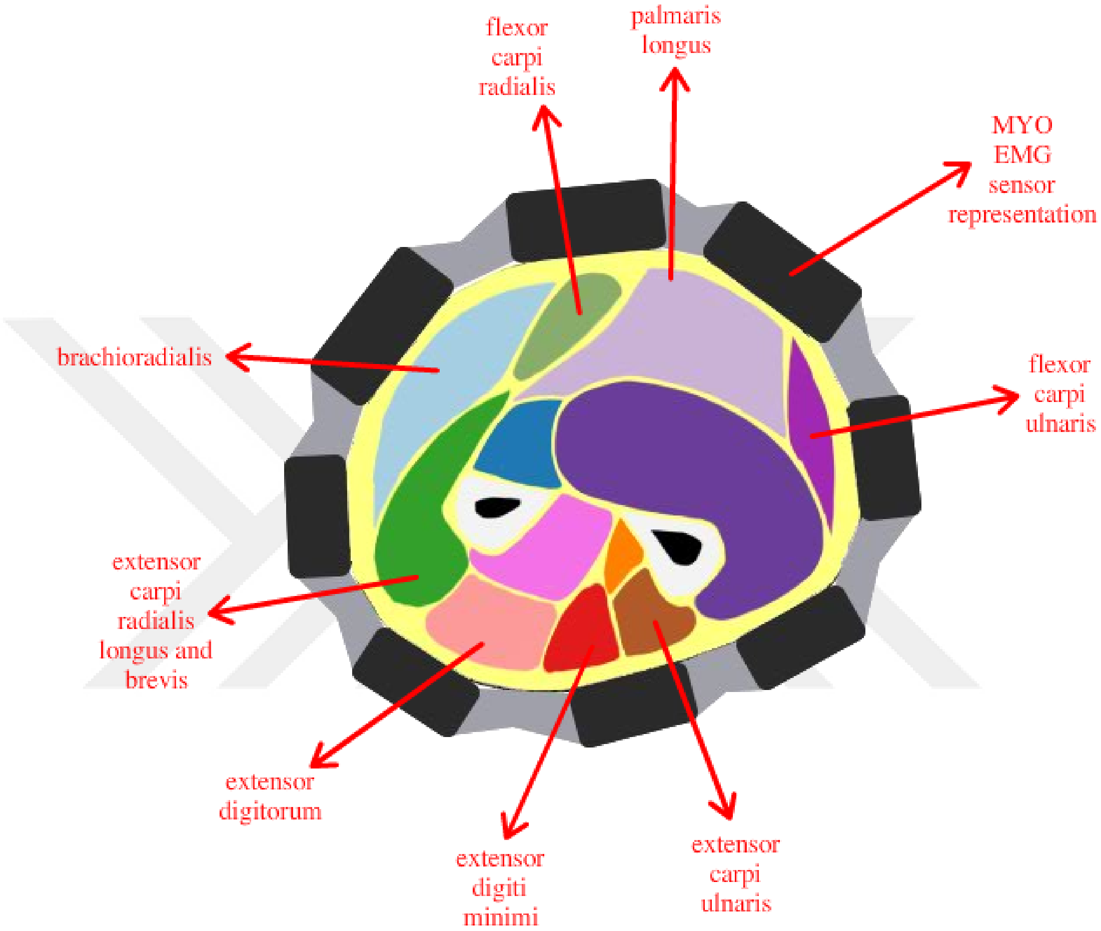


Figure 1.3: Representation of Myo on superficial forearm muscles. Adopted from "Cross section of the muscles of the distal forearm. Some extensor muscles, such as the anconeus, are not visible as they are situated proximally in the forearm." by TeachMeAnatomy.com, retrieved from http://www.bartleby.com/107/Images/large/image417 .gif Used under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license (https://creativecommons.org/ licenses/by-nc-nd/4.0/)

CHAPTER 2

BACKGROUND

In this chapter, background information on gesture recognition process is given. Gesture recognition is an application with a strong machine learning component. Hence, machine learning steps can directly be applied to gesture recognition process.

Gesture recognition process consists of 3 main steps and 2 optional steps (see Figure 2.1).

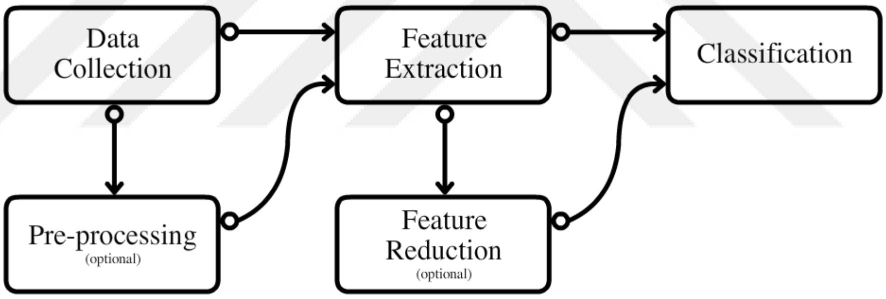


Figure 2.1: Gesture recognition process

In the data collection step, physical gestures are converted into signals that represent it. These signals are then digitized. The raw data can optionally be pre-processed. Afterwards, feature vectors that summarize the gestures are extracted from either raw data or pre-processed data. The next step is the elimination of feature vectors. For performance issues, the cardinality of feature vectors are reduced by eliminating some elements which do not provide a significant improvement and also reduce the generalization capacity of the employed machine learning algorithm. Finally, the resultant feature vectors are employed in the classification step.

In the following subsections, steps of gesture recognition process is explained in detail. These include data collection practices, pre-processing techniques, feature extraction methods and classification algorithms.

Table 2.1: Types of data for each data collection method

Methods		Data
Vision Based	Normal Camera	RGB image
	Depth Camera	Infrared image
		Depth image (depth value of each pixel)
Sensor Based	Accelerometer	3-tuple acceleration data wrt x, y and z axes
	Gyroscope	3-tuple rotation data wrt x, y and z axes (roll, pitch, yaw respectively)
	Magnetometer	3-tuple magnetic field intensity data wrt x, y and z axes
	Light Sensors	A single emitted light value for each channel
	Air Pressure Sensors	A single air pressure value for each channel
	Flex Sensors	A single flexion and abduction value for each channel
	EMG	Continuous muscle activity signal values for each channel

2.1 Data Collection

Data collection is the first phase of gesture recognition process. Gesture recognition is, in essence, the automatic labeling of signs. In order to recognize gestures, these need to be represented digitally. This representation may consist of data obtained from different sources. List of the different types of data and their corresponding sources can be seen in Table 2.1.

If the initiation and completion instances of a gesture is not to be detected but merely its presence, then each data collection session will start as the test subjects start performing each gesture and completed as the test subjects complete performing the gesture. To be more clear, steps of this type of data collection is enumerated as follows:

1. The test subject is requested to perform one of the hand gesture.
2. The test subject starts performing the gesture.
3. Data collection process is started.
4. Data collection process is finished, gesture data are stored in HDD.
5. The test subject is requested to finish performing the gesture.
6. The test subject finishes performing the gesture, release her hand to put it back to the rest position.

This type of data collection is mostly valid when hand rest position is included in the gesture set to be recognized. Classifier is always active and returns one of the gestures or hand rest position as a result.

On the other hand, if the initiation and completion instances of gesture are also of value, data collection session will start before the test subjects start performing gestures and will be completed just after the test subjects finish their performance and reach hand rest position. Steps of this type of data collection is explained as follows:

1. Data collection process is started.
2. The test subject is requested to perform one of the hand gesture.
3. The test subject starts performing the gesture.
4. The test subject is requested to finish performing the gesture.
5. The test subject finishes performing the gesture, release her hand to put it back to the rest position.
6. Data collection process is finished, gesture data are stored in HDD.

For this type of data collection, whether the test user is performing gesture or having rest is detected with different methods. For example when using EMG sensors, if the total muscle activity is less than a threshold value, then it is assumed that the test user is having rest. When the total muscle activity exceeds the threshold value, then the classifier is activated and it decides which gesture is being performed.

After data collection process is completed, collected data sets are then either pre-processed, or if a further pre-processing stage is not necessary, used in the feature extraction step.

2.2 Pre-processing

In some cases, collected raw data need to be processed prior to the feature extraction step.

Due to the dimensionality and nature of the acquired signals, pre-processing for vision and sensor based gesture recognition approaches are different. Some of the pre-processing techniques for vision based systems used in researches can be listed as follows:

- Interpreting the views and positions to create 3D model of hand from 2D images captured by different cameras [28]
- Applying Gaussian skin color model and normalized histogram on image data for skin and background segmentation [29]
- Applying discontinuity and similarity image processing algorithms on image data for hand shape segmentation [29]

- Binarizing grayscale image and applying erosion and dilation to binarized image to obtain clean binary images with required markers [23]
- Applying color filtering, erosion and smoothing techniques on images to obtain clear hand shape [30]

For sensor based systems, pre-processing techniques are applied to unidimensional signal data. Some of these techniques used in studies can be listed as:

- Applying interpolation or extrapolation and scaling to normalize each signal data to equal length and amplitude [14, 31, 32]
- Applying notch filter to eliminate the power line interference and applying low-pass filter to eliminate high frequency noise [33]
- Applying moving average filter to a signal for noise reduction [13, 34, 35, 36]
- Applying zero-mean and band-pass filters to remove the DC component and unwanted interference from signal [37]
- Applying smoothing functions for denoising [38]

As can be observed, the main objectives of pre-processing step are to normalize raw data and/or to remove/reduce unwanted noise components. Pre-processing phase generally improves accuracy at the classification step as it allows the calculation of better feature vectors in the feature extraction stage.

After the pre-processing stage, gesture recognition process continues with the feature extraction step. Next subsection describes possible features to be extracted and used in the classification step.

2.3 Feature Extraction

Information embedded in raw or pre-processed data should be represented more compactly in order for classification algorithms to be used. In general, a feature vector is calculated for a signal in short time windows, resulting in several feature vectors representing the same gesture. As long as they are relevant, the diversity of features may also increase the classification accuracy.

Some of the feature extraction techniques are summarized in the following subsections.

2.3.1 Raw or Pre-processed Data

Collected or pre-processed data can be used as a feature directly. For example, output from each EMG sensor (number of sensors on device, e.g. 8 channels), accelerometer (x, y, z axes), gyroscope (x, y, z axes) or magnetometer (x, y, z axes) can be separate feature vectors in sensor based systems. Similarly for vision based gesture recognition, RGB or HSI channels of images can be possible features vectors.

In this study, we represent a data vector as $\mathbf{x}(n)$ and its individual elements as $X_i(n)$ where n is the time index.

2.3.2 Inter-Element Difference (IED)

This feature is simply the differences of consecutive data values in each data vector. Representation can be seen as follows:

$$\mathrm{IED}_i(n) = X_i(n+1) - X_i(n) \qquad \text{for } n = 1, 2, \ldots, N-1 \tag{2.1}$$

where N is the size of each data vector and i is the vector index (e.g. EMG channel index).

2.3.3 Detected Markers

In vision based gesture recognition systems markers can be used and tracked for identifying hand parts. The spatial relationships between these markers can then be used as features. These include distances between pairs of detected marker positions or angles between pairs of marker directions with respect to a common origin. For example, assuming that fingertips and palm of the hand are detected as in [18], distances between each fingertip marker combinations can be used as separate features. Furthermore, five vectors representing each finger can be generated by connecting each fingertip marker with palm marker. Angles between those consecutive vectors can be used as four separate features as well.

2.3.4 Mean Absolute Value (MAV)

Mean absolute value (MAV) is a measure of the strength of the signal. MAV is calculated as in equation 2.2:

$$\mathrm{MAV}_i = \frac{1}{N}\sum_{p=1}^{N} |X_i(p)| \tag{2.2}$$

As an example, if MAV is applied to 8 EMG channels with i is the index of channels and N is the number of elements in each channel, then a new value set showing the average strength of the multichannel signal is obtained.

Furthermore, MAV can be used to smooth multichannel data, which results in the reduction of number of vectors. In that case, equation will become:

$$\text{MAV}(n) = \frac{1}{C}\sum_{p=1}^{C} |X_p(n)| \tag{2.3}$$

where C is the total number of channels.

MAV can also be used in vision based gesture recognition. MAV of all RGB bits of an image as well as MAV of each column or row can be used as features.

2.3.5 Moving Averaging Window (MAW)

This technique is used for smoothing the signal data. A time window $W(n)$ with a finite duration of size M is applied to the data within the window. The time window is typically symmetric around the origin:

$$\text{MAW}_i(n) = \sum_{p=n}^{n+M-1} W(n)|X_i(p)| \qquad \text{for } p = 1, 2, \ldots, (N - M + 1) \tag{2.4}$$

As it can be seen in Equation. 2.4, after calculating each MAW value, the time window is shifted by one sample in time resulting in a series of MAW values which are smoother in time.

2.3.6 Root Mean Square (RMS)

If X_i^2 is used instead of $|X_i|$ in equation 2.2, situations of whether performing a gesture (active) or having rest (idle) can be more distinguishable assuming that the signals are noisy. Roots of each value calculated in equation 2.2 with aforementioned modifications give RMS values as in equation 2.5

$$\text{RMS}_i = \sqrt{\frac{1}{N}\sum_{p=1}^{N} X_i(p)^2} \tag{2.5}$$

2.3.7 Standard Deviation (STD)

STD of a vector shows the measure of how spread out data values are. In order to calculate STD, first MAV of data vector must be obtained as in equation

2.2. Then, variance of data vector must be calculated as follows:

$$\text{Variance}_i(n) = \frac{1}{N}\sum_{p=1}^{N}(X_i(p) - X_i')^2 \tag{2.6}$$

where X_i' is the mean of the whole data or each segment. Finally, STD is the square root of variance:

$$\text{STD} = \sqrt{\text{Variance}} \tag{2.7}$$

2.3.8 Circular Harmonic Coefficients

Spherical harmonic functions are special functions defined on the surface of a sphere and widely used in indirect lighting and modeling of 3D shapes [39]. Similar to spherical harmonic functions which are defined on the sphere, circular harmonic functions are defined on the contour of a circle. Circular harmonics can be used to encode values that vary based on a single angle instead of two like in spherical harmonics [40]. Moreover, circular harmonics are very effective to recognize rotation invariant patterns.

In circular harmonic coefficient calculation, there are orthogonal and orthonormal basis functions. For this thesis, basis functions used in the study of Hsu and Arsenault [41] is used. It can be seen in Equation 2.8.

$$B_n(\theta) = e^{-in\theta} \tag{2.8}$$

which alternatively can be rewritten as

$$B_n(\theta) = cos(n\theta) - isin(n\theta) \tag{2.9}$$

where n is the order of circular harmonic function.

In order to obtain circular harmonic coefficients of a function, the function is projected onto each of the basis functions as in Equation 2.10.

$$w_n(\theta) = \int_0^{2\pi} f(\theta)B_n(\theta)\mathrm{d}\theta \tag{2.10}$$

Using these coefficient values as feature vectors is sufficient for the scope of this thesis, but further information on how to reconstruct original function by using circular harmonic functions are explained briefly for the sake of completeness.

The original function is simply reconstructed by scaling the conjugate of basis functions with corresponding weight values and adding them together as in

Equation 2.11

$$f(\theta) = \frac{1}{2\pi}\left(\sum_{n=0}^{\infty} w_n(\theta)[B_n(\theta)]^*\right) \tag{2.11}$$

After extracting features from data by using some of aforementioned techniques, they can be used in the classification process directly. Alternatively before classification, the number of feature vectors can be reduced by applying feature reduction methods some of which are described in detail in the following subsection.

2.4 Feature Reduction

Feature reduction (also known as dimensionality reduction) is one of the optional step of gesture recognition process. After features are extracted from data, this step can be performed to reduce the number of feature vectors. This step can be beneficial if the size of the data is big, because the main reason to perform feature elimination is to reduce the storage space of the data and to improve the performance of the system by reducing the execution time of the classification algorithms.

In the following subsections, some of the feature reduction methods are explained briefly.

2.4.1 Principal Component Analyses (PCA)

One of the most common statistical feature reduction technique that is widely used in vision based [42, 43, 44, 45] and sensor based [13, 32, 37, 46] gesture recognition systems is PCA. By applying PCA, feature vectors are transformed into a smaller number of linearly uncorrelated variables, called principal components. For example, combination of 2 sensor data vectors, which are extracted as 2 separate feature vectors have significant differences for different gestures. PCA method detects it and creates a new principal component with the high variance value from these 2 feature vectors.

2.4.2 Singular Value Decomposition (SVD)

Another technique that can be used for dimensionality reduction is SVD. Combination of feature vectors form a new matrix. This m-by-n matrix can be represented with dot product of 3 matrices:

- m-by-r left singular vectors of the matrix
- r-by-r diagonal matrix containing singular values of the matrix

- r-by-n right singular vectors of the matrix

If the singular values are ordered in descending order, some of the largest singular values can form a new low order representation [47].

2.4.3 Random Projection (RP)

For very high dimensional data sets, RP is more preferable than any other method [48]. k-by-N Random projection matrix is the dot product of k-by-m random matrix and m-by-N matrix composed of m dimensional original data sets [47, 49]. This method is based on Johnson–Lindenstrauss lemma [50] which explains that small set of points in high dimensional space can be represented in low dimensional space by preserving the distances between points.

2.5 Classification

Classification is the final part of gesture recognition process. Once the features have been extracted from signals or images in the feature extraction step or the number of feature vectors with redundant information have been reduced in feature reduction step, one of the classification methods or combinations of some of them are applied to sort out performed gestures.

In machine learning applications, prediction methods can be categorized in 3 main groups:

- **Supervised Learning** is a system where inputs and outputs are provided by a supervisor [51]. By using training inputs and corresponding outputs, system is trained and after that point, the trained system returns a resulting class when a new input is provided.
- **Unsupervised Learning** is a system where pattern structures and regularities are statistically found by looking at only inputs because outputs are not provided [51].
- **Reinforcement Learning** is applicable for dynamic systems where set of actions are output with respect to the created policy. In [51], Alpaydin exemplified reinforcement learning with playing a chess game where strategy and set of moves will bring victory instead of a single move.

There are plenty of learning and classification methods used in gesture recognition applications. Some of the most well-known algorithms also mentioned in the literature survey of this thesis are explained in detail in the following subsections.

2.5.1 Artificial Neural Networks (ANN)

Neuron is defined as an electrically excitable cell that processes and transmits information through electrical and chemical signals [52]. ANN is an artificial brain that is input with set of data and outputs a result. A simple representation of ANN can be seen in Figure 4.11.

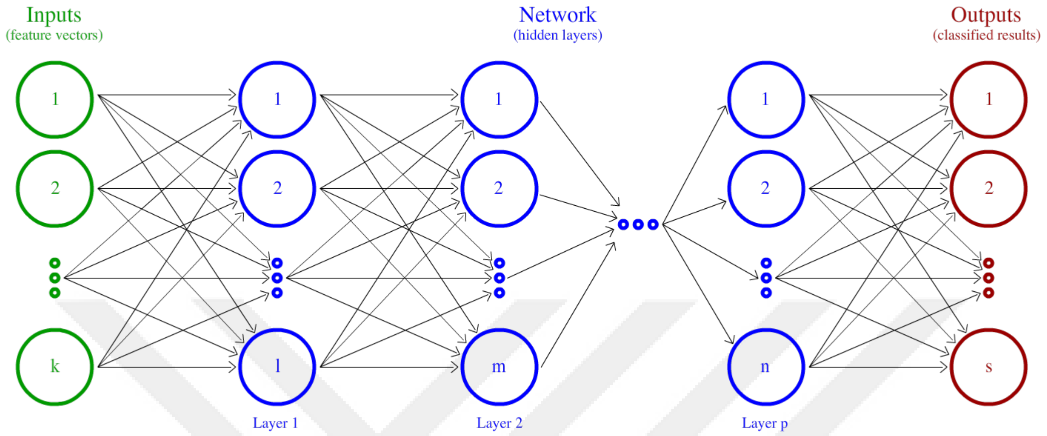


Figure 2.2: Artificial Neural Network Representation

Some part of the data is used to train empty network. Amount of the data used for training the network is significant for the accuracy of outputs.

There can be one or more hidden layers in the network and one or more neurons in each layer. There is a trade-off between accuracy and performance, hence the number of layers and neurons depends on the network model.

Each input has a value x_i and corresponding weight value w_i. Each neuron also has weight value if the network has more than one layer. Inside of each neuron, arithmetic summation which will be input of an activation function is calculated as in Equation 2.12 [53].

$$\text{net} = \sum_{i=0}^{n} x_i w_i \tag{2.12}$$

Some of the activation functions are listed in [54] as identity, logistic, hyperbolic, exponential, softmax, unit sum, square root, sine, ramp and step functions.

There are also plenty of neural network types according to the algorithms used. Five of them explained in [53] are Back-propagation Neural Network, Bayesian Neural Network, Log-linearized Gaussian Mixture Network, Recurrent Network and Wavelet Neural Network.

In the literature review chapter of this thesis, it can be seen that in [14, 18, 55, 56, 57], neural networks were used for classification process.

2.5.2 Hidden Markov Models (HMM)

HMM was defined in [58] as a finite model that describes a probability distribution over an infinite number of possible sequences. In the same research, it is also mentioned that HMM is a general statistical modeling technique for time series. Other than gesture recognition, HMMs have been widely used for speech recognition, data compression, artificial intelligence, pattern recognition, genome sequencing, image sequence modeling and object tracking [59].

HMM is an extension of Discrete Markov Processes where all events are observable. The difference is that HMM has hidden events as well as observable events. Viterbi algorithm can be used to reveal hidden event sequences by looking observed sequences [60]. This algorithm basically calculates highest probability of possible hidden state sequence on trained HMM with observations. More detailed example and visualization which is obtained from [61] can be seen in Figure 2.3 [62].

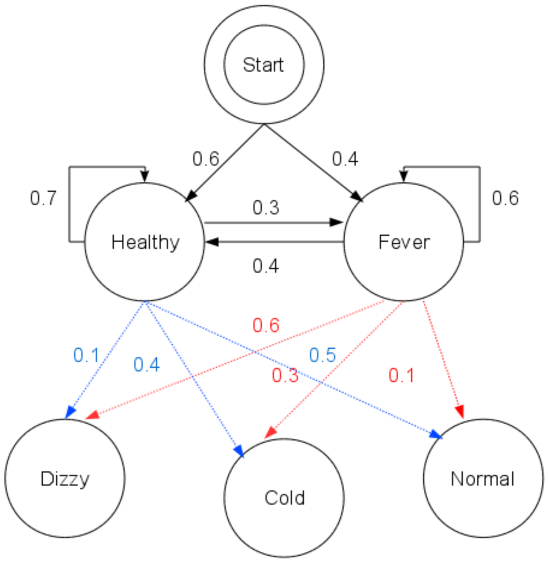


Figure 2.3: HMM Example

In this example, doctor visits by a patient are simulated. The patient can be in 3 observable states: dizzy, cold or normal. Also the patient can be in 2 hidden states: healthy or with fever. The system had been trained with some observations of the doctor and transition probabilities which can be seen in Table 2.2 and emission probabilities which can be seen in Table 2.3 were obtained. Besides, there are starting probabilities of being healthy and with fever given as 0.6 and 0.4 respectively.

The doctor observes the patient for 3 days and sees that the patient is dizzy (day 1), normal (day 2) and cold (day 3). What the algorithm finds is that whether the patient is healthy or fewer for each day.

In Figure 2.4 all possible sequences with corresponding probabilities can be seen. It is obvious that the highest probability value is 0.01344 with the se-

Table 2.2: Transition probabilities for the given example

	Transition Probabilities	
	Healthy	Fewer
Healthy	0.7	0.3
Fewer	0.4	0.6

Table 2.3: Emission probabilities for the given example

	Emission Probabilities		
	Dizzy	Cold	Normal
Healthy	0.1	0.4	0.5
Fewer	0.6	0.3	0.1

quence of fewer - healthy - healthy.

As a result of the system, if dizzy - normal - cold is inputted into the trained HMM, fewer - healthy - healthy is outputted.

Most of the studies mentioned in the literature review of this thesis use HMM for gesture classification such as [31, 38, 63, 22, 64, 65, 66, 67, 68, 69, 70].

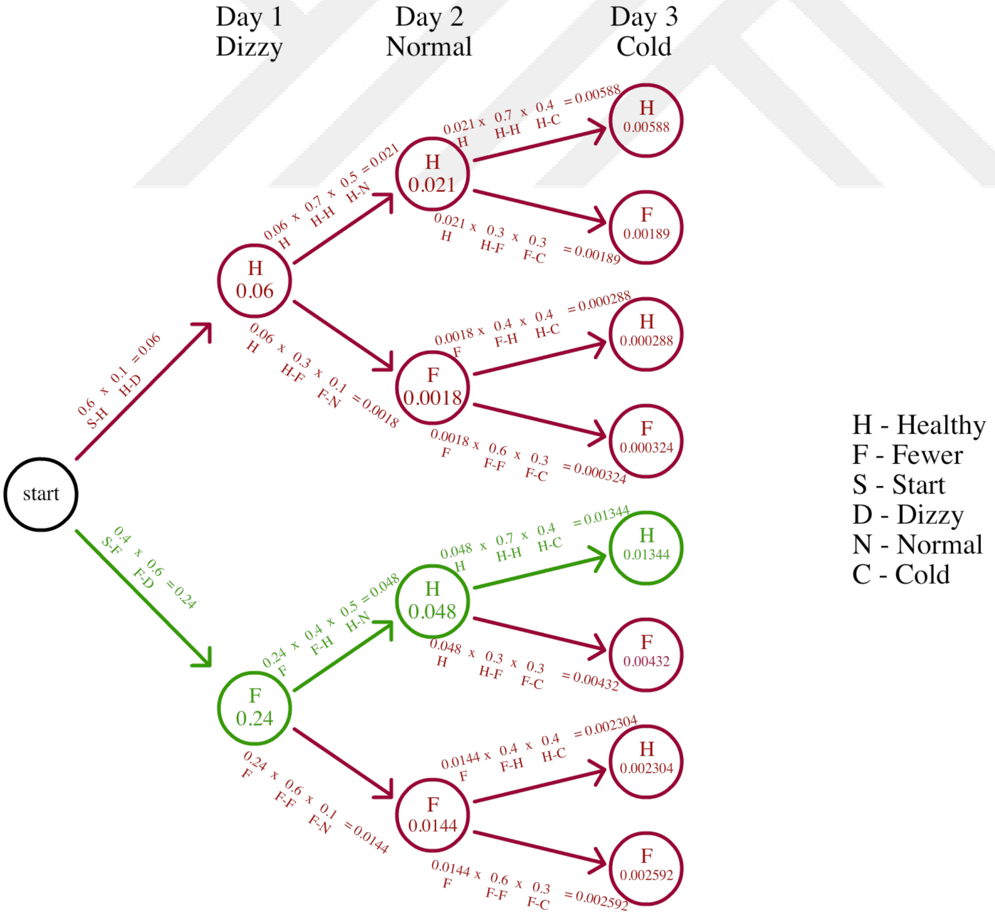


Figure 2.4: All the calculations of Viterbi algorithm for given example

2.5.3 Support Vector Machines (SVM)

Boser *et al*. first presented SVM in [71] as a training algorithm that maximizes the margin between the training patterns and the decision boundary. Firstly, data points are distributed on a system. Then, support vectors are decided. Finally, in the middle of the maximum margins between 2 classes, optimal hyperplane is chosen. From that point, the class of the new data is decided with respect to the chosen hyperplane.

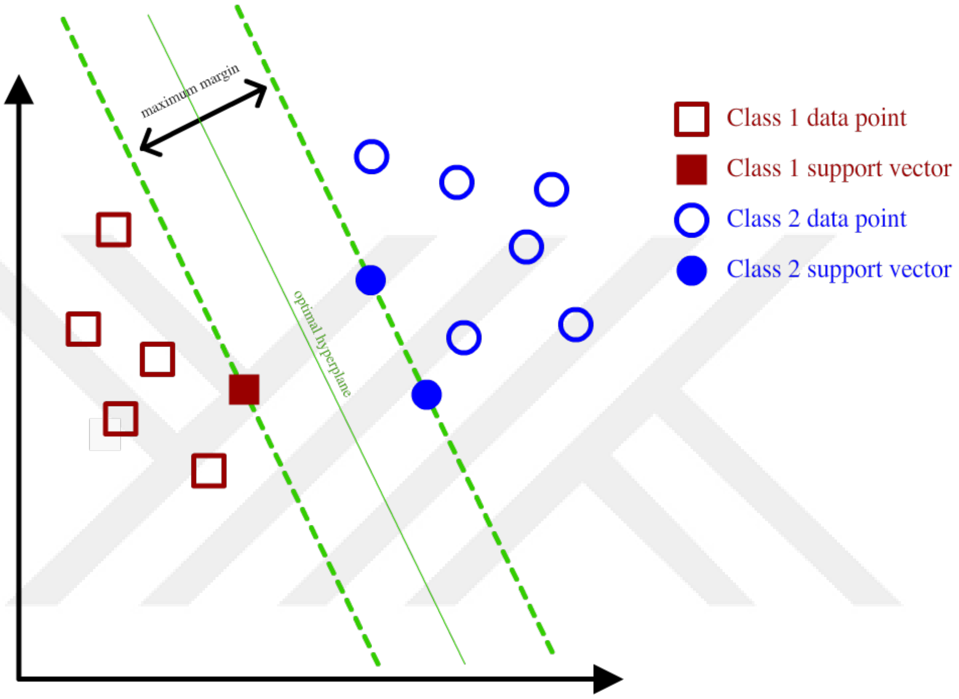


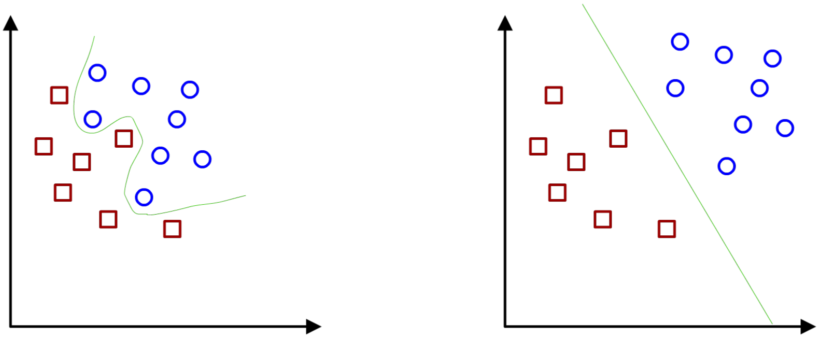
Figure 2.5: Support Vector Machine Representation

Data points are not always linearly separable as shown in Figure 2.5. Thus, a specific kernel transformation function can be applied to data to make them linearly separable. Representation can be seen in Figure 2.6.

Number of classes can be more than 2. In that case, there are 2 strategies to apply:

- **one-vs-all** is a method where one class is separated from other classes by comparing with the rest of the data. In this setup, there are 2 classes, one of which is the selected class and the other class consists of the remaining data. SVM is executed for each class separately and the highest probability result decides the class of new data.

- **one-vs-one** is a method where classification algorithm is executed between every combination of classes. The highest summation of the probability scores of each class determines the class of new data.



(a) With original data

(b) After transformation

Figure 2.6: Kernel Transformation Function Representation

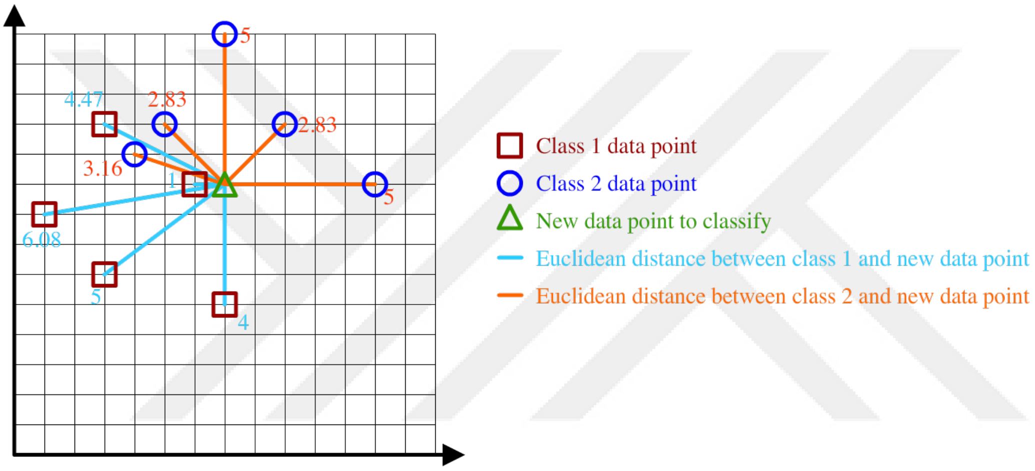

5
4.47
2.83
2.83
3.16
1
5
6.08
5
4
□ Class 1 data point
○ Class 2 data point
△ New data point to classify
— Euclidean distance between class 1 and new data point
— Euclidean distance between class 2 and new data point

Figure 2.7: K-NN Example

Some of the studies (also mentioned in the literature review chapter of this thesis) in which SVM was used for classification are [33, 35, 37, 72, 73].

2.5.4 K-Nearest Neighbor (K-NN)

K-NN is one of the easy-to-implement and low-complexity classifier algorithms [28]. Features are represented in multidimensional space and the distances between new data and training data is calculated separately. Distance algorithm can be Euclidean or Manhattan distance as well as any other custom distance calculation function. Then, the classes of K minimum distance data, in other words, the closest K data are listed. If K=1, class of the closest data is output. If K>1, majority vote decides the resulting class. At that point, different weight values can be applied for each class or data for final classification. For example, more closer data may have higher weight for more contribution of nearer neighbor.

To be more clear, a simple K-NN example is illustrated in Figure 2.7. In this illustration, 5 data points from each class are distributed over a coordinate system. When a new data point comes, its classification will be performed by calculating Euclidean distances between that data point and already classified data points. Closeness of the classified data points to the new data point is sorted in Table 2.4.

Table 2.4: Euclidean distances between classified data points and the new point

Closeness	1	2	3	4	5	6	7	8	9	10
Class	1	2	2	2	1	1	2	2	1	1
Distance	1	2.83	2.83	3.16	4	4.47	5	5	5	6.08

Outputs of the K-NN algorithm with some possible parameters are listed as follows:

- If k=1, class of the closest data point is 1. Therefore the new data point is classified as Class 1.
- If k=5, there is no weight parameter and result depends on the majority vote; the new data point is classified as Class 2. The reason is that there are 2 data points from Class 1 and 3 data points from Class 2 in the 5 closest data points.
- If k=3 and the weight parameter is assumed to be inverse operation, then the weighted effect of the closest data point is 1/1=1, 2nd closest data point is 1/2.83=0.353 and 3rd closest data point is 1/2.83=0.353. Due to the fact that total vote is calculated for each class as in Equation 2.13, the new data point is classified as Class 1 (1 > 0.353 + 0.353).
- If k=4 and the weight parameter is assumed to be inverse operation again, then the the new data point is classified as Class 2 according to the Equation 2.13 (1 < 0.353 + 0.353 + 0.316).

$$\text{Total vote for class } a = \sum_{i=1}^{N} F_{\text{weight}}(a_i) \tag{2.13}$$

where N is the number of data points in class a and F_{weight} is defined as in 2.14.

$$F_{\text{weight}}(x) = \frac{1}{x} \tag{2.14}$$

[13, 36, 56, 72, 73, 74, 21, 75, 76] are the researches reviewed in this thesis where K-NN algorithm is used in classification step of gesture recognition process.

CHAPTER 3

PREVIOUS WORK

Gesture recognition is an open area for many researchers not only there is a hype on natural interfaces with the improvements of technology, but also there are different types of powerful hardware devices which make researchers achieve high accuracy on gesture recognition easily. There are several areas using gesture recognition in HCI field since there exists many kind of devices and techniques to recognize gestures. In this thesis, these techniques are classified in 2 main groups with respect to data collection procedures : vision based gesture recognition techniques and sensor based gesture recognition techniques. This chapter discusses the literature review of all gesture recognition techniques and their usage areas.

3.1 Vision Based Gesture Recognition Techniques

Vision based gesture recognition techniques mostly use cameras to detect hand and finger shapes. This can be done in the following methods:

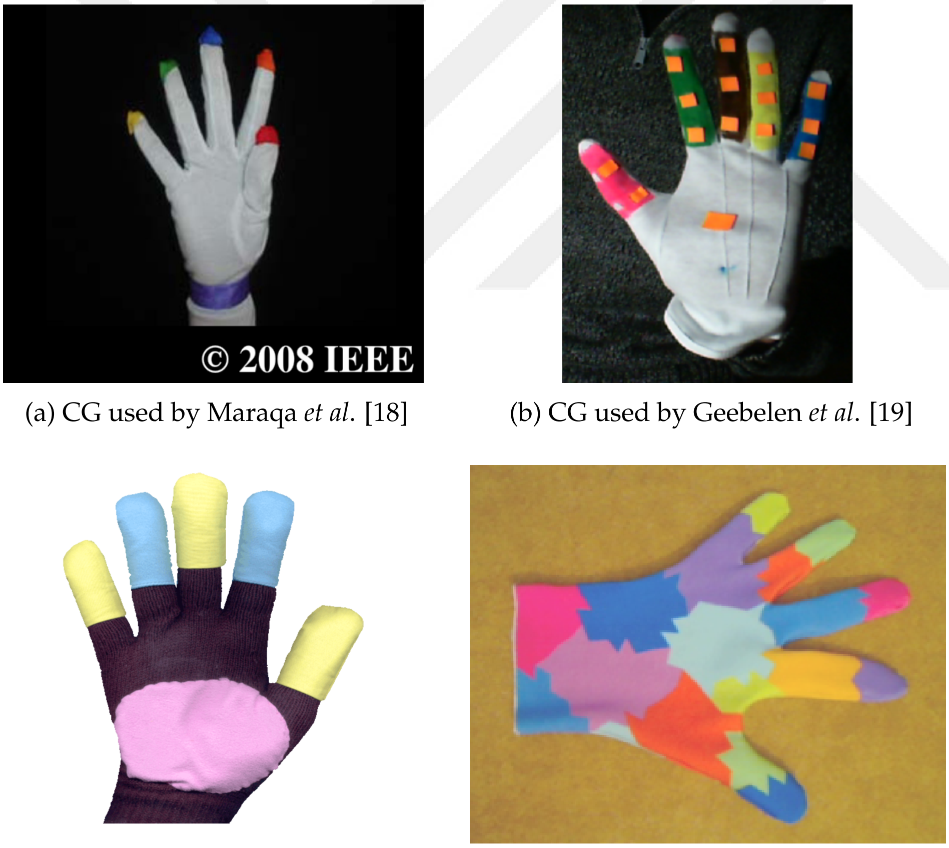
- Creating 3D model of hand
- Detecting bare hand by extracting 2D appearance of hand shape from the background
- Detecting color patterns on fingers or palm of the hand when color gloves (CG) are worn

The performance and accuracy of gesture recognition with cameras are negatively affected by illumination and lighting changes, camera resolution and movements, complicated backgrounds and depth of color. Due to the fact that these environmental aspects are influencing the gesture recognition success rate, a system may detect hand and gestures effectively in one specific environment and the accuracy may decrease when the same system is set up in a different environment. Moreover, cameras are not applicable for portable environments because they may need cumbersome setups with cables or tripods. These negative conditions make environment factor a constraint for vision based gesture recognition systems.

On the other hand, vision based techniques are affordable because no hardware equipment is needed other than cameras. Furthermore, these techniques have some advantages over other techniques. Hand shape is identical for all healthy test subjects, which makes algorithms work subject independent. Hand size or finger thickness of different subjects do not have any significant effect on gesture recognition algorithms because the fundamentals of gesture recognition algorithms are shape and appearance of hand which are identical for any healthy subject. This makes any hand detectable by a single algorithm.

3.1.1 Camera and Color Glove

Using CG is a cheap and effective solution for separation of hand parts. Fingers and palm are painted with different colors to make them distinguishable. Cameras track different colors and algorithms map each color with a specific finger or palm of the hand so that they do not have to deal with separating same color fingers from each other.



(a) CG used by Maraqa *et al.* [18]

(b) CG used by Geebelen *et al.* [19]

(c) CG used by Lamberti *et al.* [20]

(d) CG used by Wang *et al.* [21]

Figure 3.1: CG Examples

Maraqa and Abu-Zaiter [18] used simple white glove with painted fingertips of the glove with different colors (see Figure 3.1a). There is also a purple band on wrist to determine the pose of hand. Color segmentation was done with image processing algorithm using HSI color system. Feature vectors were consisting of 4 information:

- angles between fingertips themselves,
- angles between fingertips and wrist,
- distance between fingertips themselves,
- distance between fingertips and wrist.

They used recurrent neural networks for classification because of the advantages of applications requiring temporal processing. This system was developed to recognize Arabic Sign Language consisting of 28 letters.

Geebelen *et al.* [19] used similar white glove with Maraqa and Abu-Zaiter however they painted the whole fingers instead of only fingertips (see Figure 3.1b). Also, they put 3 small orange markers on each colored fingers to detect the direction of fingers and 1 small orange marker on palm to separate the palm from the fingers. For hand shape data collection and traction, they used a simple camera with a resolution of 640 x 480 pixels and 30 fps. Main goal of the research was to reconstruct 3D model of the hand and track it in real-time accurately.

Lamberti and Camastra [20] used a simple glove with only 3 colors. Pink color was used to paint the palm of the hand and yellow and blue colors were used to paint adjacent fingers as shown in Figure 3.1c. Hand shapes were collected from the camera in RGB color space and converted into HSI color space for segmentation. 5 lines were extracted from each segmented hand shape, starting from the centroid of the hand, palm, to the each fingertip. 4 angles between those lines and length of 5 lines were used as feature sets. LVQ was preferred as a classifier because it was said to have more advantages of requiring moderate computational resources. Combination of 3 different LVQ was experimented to recognize 13 different gestures and success rate close to 98% was achieved.

Wang and Popović [21] used a different type of CG which has color patterns on as shown in Figure 3.1d [77]. Patterns are different for inside and outside of the hand, which makes the hand possible to be tracked rotation independently. Their approach was slightly different from other researchers. They created a database consisting of hand poses. They collected images of hands by using a normal camera and converted them into tiny images consisting of color bits. Afterwards, these tiny images were mapped to a specific hand pose element stored in the database. This generic system was tested with 3 proof-of-concept applications. These applications are driving an animated character, manipulating rigid bodies with physics and spelling American Sign Language (ASL).

Keskin *et al.* [22] used a pure black glove to extract hand image from background easily assuming that all the experiment was performed with constant light background. They used Kalman Filter [78] for noise reduction of the captured hand images from normal camera and the gestures were classified with Hidden Markov Models (HMMs). All 8 gestures are based on tracking the same shape of the hand with different movements. The system was tested with a paint application controlled with gestures and 98,75% accuracy was achieved.

Diaz and Payandeh [23] presented hand pose and gesture estimation system based on a glove with reflective markers. They used a normal camera together with an infrared illumination source to maximize the contrast of reflective markers on the glove in order to differentiate them from background. Captured images were converted into binary images and marker shape polygons were detected with sophisticated image processing algorithms. By doing this, fingers, palm of the hand and backside of the hand could be identified. Gesture set to be recognized was chosen based on the number of fingers shown and side of the hand. State machine was selected to determine hand pose. Tracking the movement, size and direction of the hand were used to recognize performed gestures in real-time.

3.1.2 Camera and Bare Hand

Recognizing hand gestures by detecting bare hand using normal cameras does not require any other equipment for inputting the system, but advanced image processing algorithms need to be developed for hand shape extraction. It has an easy set up and is more generic for being suitable for any hand size comparing to the CG method.

Binh *et al.* [67] developed a real-time system that can recognize 36 letters of ASL with the accuracy of up to 98%. They extracted the hand from the image by tracking the skin color. Similar to Keskin *et al.* [22], they also used Kalman Filter [78] for identifying the image regions. Pseudo 2D HMM classifier was used because captured images were in 2D. Experiments showed that the system reached up to 98% accuracy in real-time.

Fang *et al.* [79] proposed a real-time robust hand gesture recognition method. They used extended version of AdaBoost method [80] for hand detection. For stable hand tracking, they used a multi-modal technique which is the combination of optical flow and color cue. Hand segmentation was performed by describing hand color in HSV color space. Then, palm and fingers were found by scale-space feature detection. 6 different gestures were tested with image browsing application by using 320 x 240 pixel normal camera and 93,8% average recognition accuracy was achieved.

Rumyantsev *et al.* [45] proposed a hand sign detection method which combines skin color detection, Principle Component Analysis (PCA) based hand gesture detection and hand centroid movement tracking. Skin color detection was performed in YCbCr and normalized RGB color spaces. Their hand detection method started with masking skin color pixels from the acquired image. It is followed with removing small and large holes in the mask. Largest areas in the mask were determined as left and right hand bounding boxes.

Lastly, found boxes were scaled to standardized dimensions to ensure size invariance. 2D image dimension was reduced to 1D vector with PCA. Weight vectors of each gesture were obtained at the end of training stage. Those vectors were compared with the weight vectors obtained by real-time captured images by calculating Euclidean distance between them to classify gestures in the test stage.

Oka *et al*. [65] used a normal camera together with an infrared camera to develop a system that can track fingertips and recognize hand gestures in real-time. With the infrared camera and an appropriate threshold value, they could detect both hands of human whose body temperature was within the range from 30°C to 34°C. After hands were detected; wrists, fingertips and palms were identified respectively by executing image processing algorithms. For classification purposes, HMM was used. They created an augmented desk interface system that lets users manipulate physical and virtual objects with hand and finger gestures. 12 gestures of the system were tested and the accuracy of single finger gestures were 99,2% while the accuracy of double finger gestures were 97,5%.

3.1.3 Depth Camera

Recently, normal cameras and infrared cameras have been widely used for data collection, until the first affordable depth camera is commercialized. After Kinect cameras made publicly available in 2010, the number of conducted researches about gesture recognition using depth cameras has been increasing. Kinect for Windows has a powerful SDK that can detect hand shape and finger joints. Researchers can use that SDK as a base for developing their gesture recognition algorithms.

Li [29] was one of the first researchers who used Kinect for creating a hand gesture recognition system. He used OpenNI Framework [81] and K-means clustering algorithm for hand detection from the data captured by the depth camera. After the detection of hand, fingers (with names) were identified by calculating distances between points on the convex hull and hand contour of the detected hand shape. Number of fingers, names of fingers and angles between fingers were used as feature vectors in classification process. Gesture recognition process started with a calibration gesture which is 5 fingers spread to identify all fingers and names of fingers in the beginning. It followed with the performing of a predetermined gesture and the system recognized that performed gesture. 2 sets of gestures were tested : 9 numbers (from 1 to 9) and 9 common body language signs (victory, okay, thumb up and down etc.). Recognition rate of the system was around 88% in average.

Frati and Prattichizzo [30] were the ones who use Kinect in early days of its release for their research on hand tracking and rendering in wearable haptics. Unlike Li, they used CLNUI Platform [82] for depth raw data, color depth data and color RGB data collection from the camera. They used the proposed system only for tracking the hand and the gesture of grasping an object.

Du and To [83] developed a system that can recognize the numbers shown with both hands individually, by using Kinect. Their system had a constraint that the person's hands should be in front of the body because they tried to extract the hand shapes by looking their depth. After the hands were extracted from the background by applying vision algorithms to the original depth image, convex and concave points of the resulting image were identified for finger detection. Those points were the elements of feature vectors and number gestures were classified by counting the number of convex and concave points.

Ren *et al*. [84] mentioned that Kinect sensor was capable of identifying human body actions, but it needed to be improved to recognize hand gestures. Therefore, they proposed a part-based hand gesture recognition system to be an alternative for this open problem. This system's constraint was to wear a black belt on the gesturing hand's wrist in order to simplify hand detection. They found the hand contour and they converted this contour into time-series by calculating the Euclidean distance between each point on the contour and the center of the hand shape. Then, they applied an improved template matching method, Finger-Earth Moving Distance, for classification of gestures. Their system was tested with 10 subjects performing 10 gestures (numbers from 1 to 10) in a challenging environment to show that the system was robust to cluttered backgrounds. Furthermore, all time-series representations of the same gestures with different rotations were similar to each other because starting points of the obtained hand shape contours also rotated with the hand. This led the system to be rotation invariant. Developed system was integrated with arithmetic computation and rock-paper-scissors applications in order to show that the system is applicable to real-life applications.

3.2 Sensor Based Gesture Recognition Techniques

3.2.1 Accelerometer Sensors

With accelerometer sensors, instead of gestures created by forming hand and fingers in different shapes, gestures consisting of movements of hands or arms can be recognized. Accelerometer sensors can be placed in armband worn on an arm, they can be embedded into smart phones or they can be a part of hand-held devices like Wii remote [85] by Nintendo or PlayStation Move [86] by Sony. Wii remote has infrared sensors while Playstation Move has a light ball, which are tracked by the cameras of their own systems in order to combine the vision information with the data coming from their accelerometer for accurate acceleration calculation.

Kela *et al*. [31] developed an acceleration based gesture recognition system to use in designing applications. They applied a questionnaire to choose the most natural gestures for controlling VCR device. After specifying 8 gestures (Right Down-Left Down, square, left, right, up, down, Clockwise (CW) circle and Counter Clockwise (CCW) circle) 3-axis acceleration data were collected

(a) Wii Remote (b) PlayStation Move (c) SoapBox

Figure 3.2: Accelerometer Device Examples

by using a small sensor device called SoapBox [87]. Firstly, they pre-processed the collected data to normalize the variation in gesture speed. Secondly, they reduced 3D data to 1D by applying vector quantisation with K-means algorithm. Lastly, they used HMM for classification of the gestures. They could reach up to 98,9% success rate by increasing the number of training vectors.

Liu *et al*. [88] used Wii remote to collect acceleration data for their gesture recognition system, uWave. The system converted the collected accelerometer signal data into time series. With dynamic time warping method, converted time series were compared with template time series for gesture matching purposes. Updating the templates with input data made the system dynamic. They preferred to choose 8 gestures from previous research because their aim was to be able to use the system in real-life applications and these gestures were the most preferred ones to use with home appliance. They collected data from subjects in different sessions through different days. The same day data were compared with all days data in terms of the gesture recognition accuracy. The developed system was integrated with a gesture based user authentication application and mobile user interface application.

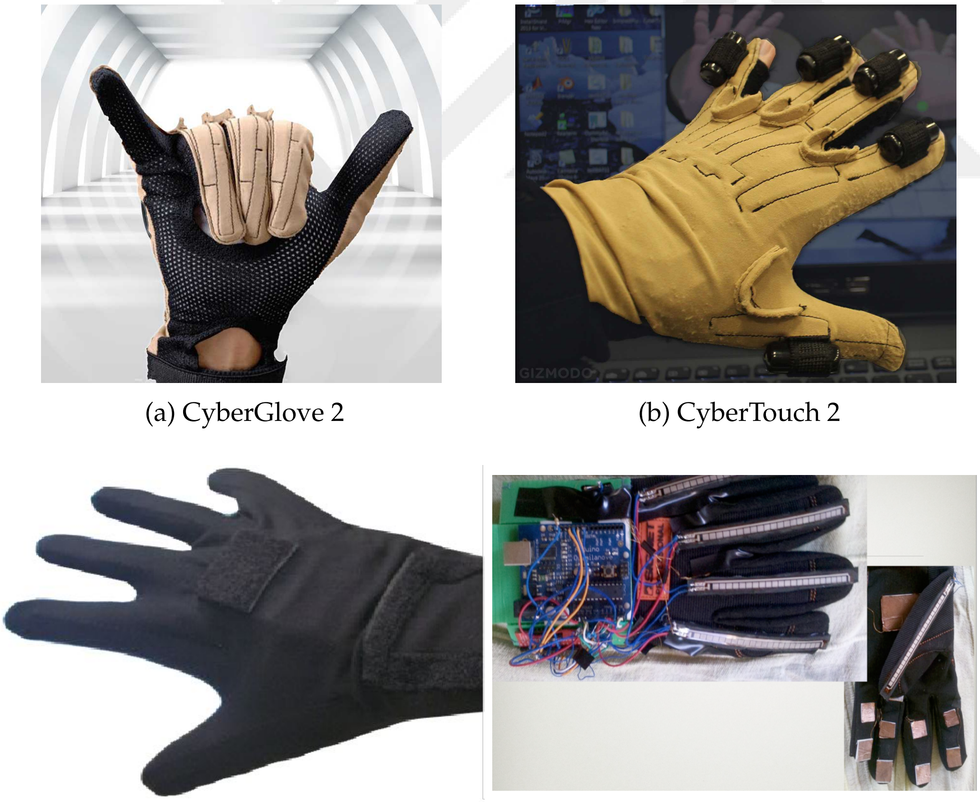
Another utilized custom hand-held accelerometer device came from Liu *et al*. [38]. They used the device to collect data for their proposal of accelerometer-based gesture recognition algorithm. In their method, 3-axis data was pre-processed for quantification and clustered with k-medoids algorithm for being robust to noisy data. Discrete HMM was used for classification. Proved with more than 94% average accuracy, the system was applied to a virtual reality application where users interact with objects in 3D environment.

Akl *et al*. [48] used random projection method to improve other researches about accelerometer-based gesture recognition. Combination of Dynamic Time Warping (DTW) and affinity propagation was used for clustering in the training stage. Test data and exemplars (the output of the training stage) were pre-processed with DTW again and random projection matrix was generated. By using this matrix, the problem was formulated as $\ell 1$-minimization problem and the solution of the problem gave the recognized gesture. With the proposed system, almost perfect accuracy was achieved in user-dependent tests and the success rate of 94,6% was reached in user-independent tests. Results seem to make the proposed method novel.

Wearable devices are becoming more popular these days and smartwatches are the latest products of this popularity. These devices are equipped with several sensors to use mainly for tracking the movements, geographic location or sportive activities of their users. Wen *et al.* [73] used accelerometer, gyroscope and linear accelerometer sensors of a smartwatch and presented a unique system that recognizes 5 fine-motor finger gestures (pinching, tapping, squeezing, waving and rubbing) to control the smartwatch itself. Raw data and distances between each axes for each sensor formed the time-series data set of the recognition system. 10 lower of 25 power bands produced by taking FFT of calculated statistical features of obtained time-series data set were decided to be feature vectors. For classification, they implemented 4 different classifier : Support Vector Machines (SVM), Naive Bayes Classifier, Logistic Regression and K-Nearest Neighbor (K-NN). The system was designed to choose the best one according to the user. 83% accuracy was reached in overall performance.

3.2.2 Flex Sensors

In gesture recognition studies, flex sensors are mostly used to detect finger movements. These sensors are commonly located on the finger parts of special data gloves. Each sensor converts the flexion and abduction of corresponding finger to digital data.



(a) CyberGlove 2

(b) CyberTouch 2

(c) DG5 VHand 2.0

(d) JhaneGlove

Figure 3.3: Glove Examples Having Flex Sensors

Liang and Ouhyoıng [64] presented a real-time continuous gesture recognition system that can output 250 vocabularies of Taiwanese Sign Language by recognizing 51 hand gestures, 8 hand movements and 6 gesture orientations. Detection of the starting and the end points of gesture was a problem because the system was input continuously. They overcame this problem by defining a time-varying parameter and used this as a threshold value. They used flexion data of 10 finger joints reported by DataGlove^{TM} as posture features; elevation data coming from the 3D tracker as position features; azimuth and roll data obtained from a 3D tracker as orientation features; the normalized 10 connected vectors of motion trajectory as motion features. These four feature sets are input to HMM for classification. They tested the recognition rate of the system in three groups: isolated gestures, short sentences and long sentences. Results showed that the success rate was decreasing down to 70% with the increase in the number of vocabularies.

Sign language recognition systems are the most common applications of gesture recognition research. Vamplew [56] proposed a system called SLARTI to identify Australian sign language gestures. Data collection was similar to the previous study except that in this study CyberGlove with 18 flex sensors was used. Moreover, features were selected the same as the previous research, they only were pre-processed with different algorithms. For this system in the classification stage, feature vectors were fed to neural network with different properties. For each feature, the network which gave the best results was chosen for the final system. The difference of this system was to have one more classification step because each Australian Sign Language dictionary element consists of sequences which start with a starting gesture, are followed by a motion of hand and finish with end gesture. 7 features were generated for each sequence. Nearest-neighbor look-up algorithm with simple distance measure and heuristic distance measure were applied for the final classification. 90% average accuracy was achieved.

One of the beneficial comparison of classification methods was studied by Parvini *et al*. [55]. They developed a gesture recognition system and compared single layer neural networks, multiple layer neural networks and GRUBC (gesture recognition by utilizing bio-mechanical characteristics) with different numbers of training, validation and test sets. Gesture set was constructed with 22 letters of ASL. CyberGlove was used in this study and the data collected from its flex sensor channels were used as feature vectors. Results of the single layer neural network with the configuration of one set for test and the remaining sets for training were close to the results of GRUBC method with the similar configuration. These methods had the highest accuracy among others.

Kumar *et al*. [75] created a hand gesture recognizer to use in an interesting application, air writing and sketching. They used DG5 VHand data glove whose flex sensors are analog. Hence, they had to implement an analog-to-digital converter. Output of this converter was used as features of K-NN classifier. Their algorithm for hand pose detection did not require any signal processing, they only checked whether the value of any specified flex sensor was between 2 threshold values. The gesture set recognized by the system contained 6 gestures (idle, left click, right click, dragging, rotating, pointing).

Camastra and Felice [76] also used DG5 VHand data glove in their gesture recognition research. In this study, Camastra preferred to use a data glove instead of a CG which had been used in his previous study [20]. For feature set, they used the data obtained from the data glove by applying bandpass filter. LVQ with 3 different learning techniques were chosen for classification process. Combination of the different LVQ classifiers which were giving the best results was used for final results. The proposed system recognized 13 gestures with the accuracy of 99,31%, which was ≈7% better according to their comparison with K-NN classifier. The system was used in VR course to control a presentation program.

Contribution of Katzenelson and Karsenty [14] to the hearing/speaking disability research area is important. They described a smart glove having name as JhaneGlove which is responsible for converting hand gestures to speech. JhaneGLove consists of 5 flex sensors for each finger, 8 contact pad sensors to detect contacts between fingers and the hand palm, accelerometer and gyroscope for motion and orientation of the hand. They designed the system as a generic gesture recognizer to be able to identify custom gesture sets with a simple calibration and training process. Neural network with back propagation algorithm was used as a classifier. The system was tested with 15, 20 and 30 signs separately and the accuracy dropped down from 92% to 80% as the number of signs increased. Furthermore, they stated that their system would need to be improved because natural sign language was much faster without the glove although they achieved to recognize each gesture in ≈1 second.

3.2.3 EMG Sensors

Electromyograph is a device recording electrical activities produced by muscles. Hand and finger gestures are activating forearm muscles when they are performed. Hence, EMG sensors can be used to identify performed gestures as they are activating different muscle groups with different frequencies. EMG sensors can be used alone to recognize static hand poses as well as they can be combined with accelerometer or IMU sensors to detect movements of hands or arms.

3.2.3.1 Only EMG Sensors

One of the first research about using EMG signals for HCI was presented by Wheeler [66] in 2003. two separate gesture sets were used for this study : virtual joystick set consisting of left, right, up and down gestures; virtual numeric pad set consisting of numbers from 0 to 9 and enter keyboard button. Number of electrode pairs and their locations were specified with respect to the gesture sets. For the first gesture set, dry electrode pairs were used while wet electrode pairs were preferred for the second gesture set. When the electrode pairs were wet, quality of the signal was higher and noise of the signal were less whereas placement of wet electrode pairs was possible to change unintentionally for each session. According to their tests, minor displacements like 1-3 mm had no effect where major displacements like 1-2 cm ruined the success rate, which proves how sensibly EMG electrode pairs have to be placed. Data collected from electrode pairs were digitized and pre-processed with Bessel Filter [89]. One of the simplest feature space, moving averages on time series representation of the signal, was selected for feature extraction. Left to right HMM with Standard Baum-Welch training algorithm was used for the pattern recognition. Results showed that EMG signals can be used to control devices. However, in order to integrate such systems to our daily life, this cumbersome process should be overcome by developing portable EMG devices.

Another study about gesture recognition by processing surface EMG signals was performed by Kosmidou *et al*. [90]. They collected 2-channel EMG data with the frequency of 1 kHz. First, they defined 16 features in either time or frequency space. Then, they decreased the number to 11 by applying discriminant analysis based on the criterion of Mahalanobis distance. Lastly, they identified 8 discriminant functions to represent those 11 features. Resulting 8 discriminant functions could classify 9 ASL gestures with the success rate of 99,4%. Moreover, 2 of 11 features were eliminated because of their low white noise tolerance. Classification with the updated discriminant functions resulted in 97,7% accuracy.

A real life application that controls an RC car with hand gestures was developed for the experimental part of the study of Kim *et al*. [36]. The challenging part of this research was that they used only 1-channel EMG sensor to obtain muscle activity signals from different muscle groups which are responsible for wrist and finger movements. They explained that the obtained signal had minimum noise whereas it had an unstable baseline. Hence, they detrended the raw signal data with a simple averaging function in pre-processing stage. They faced detecting end of the pattern problem and they overcame it by applying RMS function. For feature sets, not only time domain features like mean value, variance and signal length were used; but also frequency domain features like FFT and Fourier variance were included. The combination of K-NN and Bayes classifiers with a decision tree was used to identify 4 simple gestures : left, right, forward, backward. In order to specify each threshold values used in feature extraction and classification stages, calibration process had to be performed for each user. Although they provided general threshold values, the system showed better performance with calibrated values because they also experienced that EMG signals would vary from person to person. Average success rate of 94% was achieved. They also compared the results by grouping the test subjects according to their gender, height and weight.

A custom, low-cost, wearable and flexible EMG device with on board processing capabilities was proposed by Benatti *et al*. [33]. They collected analog data by using electrode pairs and applied notch filter for removing power line interference and low pass filter for noise reduction. The composition of 8-channel pre-processed data was used as feature vectors. For classification, they chose SVM because their preference was better accuracy more than computational cost. 123 support vectors were trained with 10% of the collected data. The gesture set they tested consisted of 6 gestures (power grip, precision grasp, open hand, pointed index and the flexion/extension of the wrist) and a hand rest position. 89,2% classification rate was achieved. Furthermore, the accuracy results of different number of input channels and gestures were compared. Moreover, the same developed algorithm was executed with the inputs obtained from Myo device and similar accuracy rates were achieved. They finished their comparison by stating that their device had advantages in terms of processing capabilities while Myo has advantages for being more portable and user friendly.

Savur and Sahin [91] used an advanced EMG device to collect data for their ASL recognition system. Ten time domain features were extracted from 8-channel, band-pass and notch filtered EMG data. These features were : MAV, modified MAV, simple square integral, root mean square, log detector, average amplitude change, maximum fractal length, minimum value, maximum value and standard deviation. One-vs-all SVM with Radial Basis Function (RBF) kernels was used for classification. They found the best sigma value which gave test and cross validation results closest to each other by testing different sigma values of kernel function. Their system identified 26 ASL letters with the accuracy of 91,73% in offline and 82,3% in real-time. Results seem to be pretty accurate, but their experiment was performed by only one test subject.

3.2.3.2 EMG + Accelerometer Sensors

Detecting only pose of hand or fingers is applicable with EMG data alone. Most of the sign language letters can be represented with static gestures. However, some letters or other applications necessitate the movement of the hand too. For detecting dynamic gestures, only EMG data are not sufficient. Therefore, other additional sensors are combined with EMG sensors to obtain more features containing the hand motion. Accelerometers are commonly used for this purpose.

Application areas of EMG + Accelerometer combination are not limited to sign language recognition. Zang *et al.* [68] proposed a system that recognizes 3 hand pose (Little finger, thumb and fist) with 6 different circular movement (CW and CCW with respect to x, y and z axis). Controlling the virtual Rubik's Cube [92] with hand gestures is the experiment application of the system. As mentioned earlier, hand poses were detected with EMG signals and hand movements were detected with accelerometer sensors in this study. Starting and end of gestures were identified by looking at the average of squared multiple channel EMG signals for being below or above of the predefined threshold values. They labeled the areas between those identified points as active areas to be used in gesture recognition process. Four-channel EMG device and a standalone 3D accelerometer were used in the system. Results of the combined system in offline was satisfactory because the number of gestures in hand pose gesture set was limited to only 3 and very distinct gestures from each other in terms of the activation of different muscle groups. Moreover, the hand movement they chose was a single circular movement in different axis and directions, which made them easier to be recognized in comparison to the more complex hand movements. Real-time test results were also successful with the accuracy of 91,7%.

Two years later from their publication, Zang *et al.* upgraded their EMG data collector device with 5-channel EMG sensors in order to cover the activities of 5 different forearm muscle groups: *extensor digiti minimi, palmaris longus, extensor carpi ulnaris, extensor carpi radialis, and brachioradialis* [70]. In this study, they used multi-stream HMM classifiers together with a decision tree for reducing the number of possible results, which showed great improvements in terms of recognition time. The decision tree decides whether the gesture is static or dynamic, whether the duration of the gesture is short or long, whether the palm of the hand is looking up or down. As a result, with the less number of HMM classifiers, more accurate and faster gesture recognition was achieved. With the proposed system, they tried to identify 72 Chinese Sign Language words and 40 sentences formed by using some of these words. More than 95% average recognition success rate was achieved. They also proved the improvement on their previous study by testing the new system with Rubik's Cube application again. Average accuracy values of 97,6% for user-specific experiments and 90,2% for user-independent were achieved.

The importance of gesture selection was studied in the research of Kosmidou and Hadjileontiadis [93]. They proposed a gesture recognition system to identify 60 Greek Sign Language words. They broke down each word into small parts in order to represent words with the sequences of movements. They categorized 5 movements (extension, flexion, left rotation, right rotation and pause) of three body parts (arm, fingers, hand) and each of the movement of the body part was symbolized with a letter. Hence, they would be able to match each gesture with the sequence of those letters. It was the reason to obtain the similarities between word pairs. In order to recognize those 60 words, they analyzed 5-channel EMG data together with 3D accelerometer data by using intrinsic mode entropy. They stated that the number of EMG channels were optimal because according to the experiments, the increase in the accuracy was not worth to increase the number of EMG channels where the decrease of success rate was significant when less number of EMG channels were used. Discriminant analysis based on Mahalanobis distance criterion was used for classification. More than 93% accuracy was achieved with the proposed system.

When the number of gestures to be recognized is high, different techniques should be used to achieve high accuracy. Li *et al.* [63] proposed a system whose two accelerometers and 8 EMG sensors were placed on both left and right forearms. Their gesture recognizer was capable of identifying 121 Chinese Sign Language subwords most of which must be performed by using both hands. They used normalized 3-axis accelerometer data, Autoregressive coefficients obtained by Hamming Window filtered EMG data and MAV of EMG channels as feature vectors. For less computational complexity and reduced recognition time, they used a decision tree combined with multi-stream HMM for classification. Average results of their tests yielded to 95,78%

success rate.

Lu *et al*. [34] put gesture recognition applications one step further by moving the system to mobile devices. While there may occur user experience problems, they presented a system that recognized nineteen gestures to control mobile phone. They defined four of them (bending wrist to right or left, spreading fingers and fist) as small-scale gestures which are different forms of hand. The remaining fifteen gestures were defined as large-scale gestures which can be performed by moving the hand performing a specific small-scale gesture to different directions. Five of them represents the arrow keys and centering, ten of them represents numbers from 0 to 9. They classified small and large scale gestures separately. For small-scale gestures, Bayes linear classifier was used where for large-scale gestures, DTW was preferred than HMM in order to avoid latency. 95% average accuracy for user-dependent and 89,6% average accuracy for user-independent results each was reported within 300 ms showed the success of the proposed system. Moreover, a questionnaire applied on participants showed that controlling mobile phones with hand and finger gestures recognized by the proposed system is accurate, practical, enjoyable, natural and controller device needs improvement to be more comfortable.

3.2.3.3 EMG + IMU Sensors

Georgi *et al*. [94] used EMG and IMU sensor combination in their gesture recognition study. Their main purpose of using IMU sensors was to generate more features by using gyroscope sensors. They wanted to create a generic system that tried to identify 12 gestures, session and person independent. Results of their study showed that creating a person-independent generic system with only EMG sensors was not possible with their technique because they do not place EMG sensors precisely on muscles. They preferred to place them in a regular pattern to make the hardware easily wearable like armbands. However, with the contribution of IMU sensors, mean accuracy of 74,3% was achieved, which is significant for person-independent recognition techniques.

3.2.3.4 Myo

Myo is a commercial device containing EMG and IMU sensors similar to the hardware used in studies reviewed in previous section. There has been plenty of research using Myo for data collection in recent years. Thus, it will be reviewed in separate subsection.

Savur is one of the first researchers who used Myo to collect data for his study [37]. Experiments of his previous research [91] was performed by using a custom EMG device as mentioned earlier. With this study, he was able to compare the results of using these two devices. The custom device was less comfortable to wear because electrode pads were sticky and EMG electrodes were connected to main unit with cables although data transmission was provided via Bluetooth. On the other hand, EMG electrode pads of the custom device could be placed on muscles sensitively, which resulted in more accurate data collection than Myo.

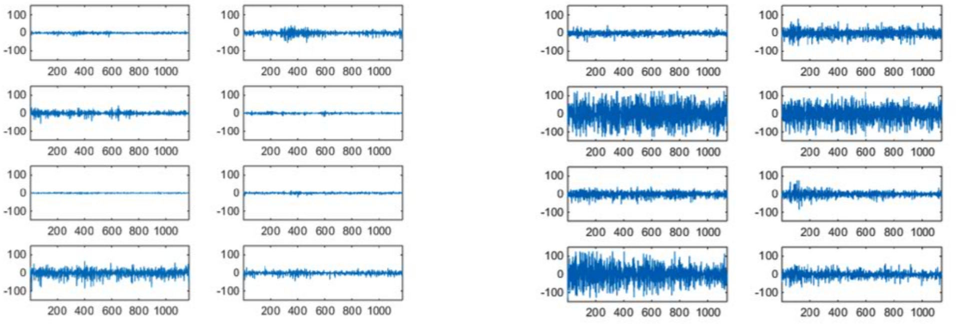
Sathiyanarayanan *et al*. [12] used Myo in their research for controlling an unmanned ground vehicle. However, they only used gyroscope of Myo for giving very simple 5 commands (up, down, left, right and halt) to the vehicle. Instead of using a single gyroscope or IMU sensor, they benefited from Myo's portability by disregarding its EMG data collection feature.

Lee and Rao proposed a procedure of collecting data using Myo in their research [95]. Their method only included the data structure to hold the collected data in an order. Although they did not give any details on machine learning classifier techniques, they tested their system with Naïve Bayes classifier. Success rate of 70% for one specific test user was achieved for recognizing 8 military gestures.

Boyali *et al*. brought a new point of view to the structure of collected data for gesture recognition in their research [96, 97]. Instead of recording data for each gesture separately, they tried to record the data of gesture pairs which were performed continuously. Their experiment subjects were switching between two gestures during the data collection process for each gesture pair. On collected gesture pairs, ordered subspace clustering based on sparse subspace clustering was used. Collaborative representation based classification was able to identify 6 gestures (predefined Myo gestures + hand relax position) with the accuracy over 97%.

Abreu *et al*. proposed a system to recognize some letters of Brazilian Sign Language [35]. They excluded the letters whose gesture performance require hand movements and included remaining 20 letters with stationary gestures to form the gesture set of their research. Collected data was pre-processed with inverting and smoothing algorithms. Then, binary SVMs were trained with one-vs-all strategy for classification process. In this process, they tried different values for RBF kernel parameters and chose the ones giving best accuracy results which is more than 98%. The most valuable contribution of their study was about performing gestures during data collection process. They tested to perform the same gestures by applying different strengths with fingers and found out that the EMG readings would vary for the same gestures. Two different data sets for the same gesture can be compared from Figure 3.4. These differences proved that performing gestures by applying similar strength levels during the data collection is as important for better accuracy as precise placement of EMG sensors on the skin.

Sánchez compared different pre-processing and classification (Decision Trees, Random Forests, Error Correcting Output Codes and K-NN) algorithms in his thesis study [13]. 7 different methods (Integrated EMG, MAV, SSI, Variance of EMG, RMS, Waveform Length, Zero Crossing) were applied for feature extraction. Sliding window filter and majority vote methods were applied to the classified data to obtain smoother results. Gesture set he used was composed



(a) Gesture is performed naturally

(b) Gesture is performed with extra strength

Figure 3.4: EMG reading differences of the same gesture by applying different levels of strength while performing gestures

Table 3.1: Active EMG sensors of Myo for each gesture

	EMG Sensor Indexes							
	1	2	3	4	5	6	7	8
Wave Left	X					X	X	X
Wave Right			X	X	X			
Fist		X			X	X	X	X
Fingers Spread		X	X					
Double Tap		X	X			X	X	

of 4 predefined Myo gestures (fist, wave right, wave left and spread fingers), 4 additional gestures (pinch, elder, voor and tiger) and 1 hand rest position. Effects of collecting data with 2 different frequencies (10 Hz and 200 Hz) also compared.

A contribution using Myo for medical researches was done by Abduo and Galster. They implemented an interface to visualize and collect real-time data in order to use for medical applications [98]. Main purpose of their study was to find a new way to recover the lost abilities of hand amputees. Gesture set used in this research was composed of finger and wrist movements, hand postures and grasping of different objects.

Similar research for hand amputees was conducted by Ganiev *et al.* [99] to control virtual robotic arm using Myo. Main topic of this study was not gesture recognition, instead they would like to show that all sensors and features of Myo were capable of being used as virtual arm. They used IMU data for representing movements and orientation of the virtual arm and predefined gestures detected by Myo's system in real-time for actions such as grasping (fist) or bursting balloons (double-tap). Important part of their study is that, they specified which EMG sensors (out of 8 sensors) were active while performing each gesture. Their specification can be seen in Table 3.1. This information would be used in classification algorithms for gesture recognition.

Usability of Myo for medical research was proven by Sathiyanarayanan and Rajan [100]. They applied System Usability Scale (SUS) questionnaire to 24 medical students to quantify the experience of using Myo for physiotherapy. Average SUS score of $\approx 70\%$ was achieved, which is mentioned to be acceptable.

Yeh *et al.* proposed a recognition system which can detect basketball referee gestures [101]. Their first gesture set consists of 9 dynamic gestures, which means they used EMG signals together with ACC signals collected using Myo. They stated that using only EMG sensors for person independent gesture recognition systems is challenging for two reasons:

- EMG signals can vary significantly from person to person.
- EMG sensors should be located precisely on muscles to obtain similar signal patterns for the same gesture performances.

In the pre-processing step, they used removing the mean, rectification and smoothing techniques. For feature extraction methods, they used DBN (Deep Belief Network) with 4 RBMs (Restricted Boltzmann Machines) and autoregressive model coefficients. For classification, they tested HMM, SVM and ANN. SVM was found to have the best classification ability among others. Their proposed system achieved 97.9% accuracy with 5-fold cross validation technique, 90.5% accuracy with leave-one-participant-out cross validation technique. Their second gesture set consists of 5 stationary gestures. These gestures are performed by fingers to show the numbers from 1 to 5. The proposed system classified these 5 gestures with the accuracy of 54.3% because only EMG data was used in the gesture recognition process.

Lastly, the study of Nymoen *et al.* [102] was about using Myo to control musical instruments with gestures. They did not implement a gesture recognizer, instead they used predefined actions that Myo recognizes by itself. In their research, they calculated the noise levels of acceleration and EMG data, rotational drift of IMU data and time lag of data streams so that Myo was decided to be eligible for controlling musical instruments. Results of their user testing showed that misclassification problems occurred due to the fact that each test user had different arm thickness. Nevertheless, test users were observed to have fun using the system. The requirement of the precision while controlling musical instruments was also mentioned to be significant.

These studies prove that Myo can be used as a data collection device for gesture recognition. Its advantages can be listed as being portable, affordable, powerful with 8-channel EMG and IMU sensors. There are also disadvantages that can be summarized as precision problems while placing EMG sensors, signal strength problems comparing to the devices use wet electrodes, low EMG frequency comparing to the advanced devices and lack of embedded computing capabilities comparing to the device used in [33].

By looking at all these researches and their application areas, it can be inferred that gesture recognition is widely used for identifying sign language letters or words. Moreover, some examples on medical treatment for the ones with physiological disabilities can be seen. Apart from these beneficial real life examples, it can also be used just for fun as a controller for physical robots, toys or abstract characters in video games.

Another fun application area of gesture recognition consists of virtual classical music orchestra conductor applications. In this type of applications, users perform gestures to control musicians playing their instruments.

In 1991, Morita *et al.* laid the foundations of the first virtual orchestra application. They implemented a computer music system which followed a human orchestra conductor's movements [103]. A conductor generally uses his dominant hand to perform the tempo of the music by using a baton, and his non-dominant hand to perform expressive gestures like crescendo, pianissimo or forte [104] in order to operate musicians with the dynamics of the music. In their research, they used infrared cameras to track the dominant hand for tempo, and data glove to collect data from the non-dominant hand for detecting gestures. The data glove used in this study has optical fiber cables located on fingers and a magnetic position sensor which detects the movements of hand and finger joints. For the non-dominant hand expressive gestures, 11 features they extracted were composed of the horizontal and vertical positions and velocities of hand; crooking of thumb, index and middle fingers; rotation and direction of palm. Their application was tested by conductors and musicians and they collected several feedbacks on tempo prediction and compensation.

Furthermore, there are some studies on virtual orchestra conductor applications like [105, 106, 107, 108]. In those researches, only the algorithms that predicted the tempo of the music by using dominant hand were studied.

Recognition of the non-dominant hand expressive gestures in a virtual classical music orchestra conductor application is the main topic of this thesis. Methods applied to identify expressive gestures of conductors are explained in the next chapter.

CHAPTER 4

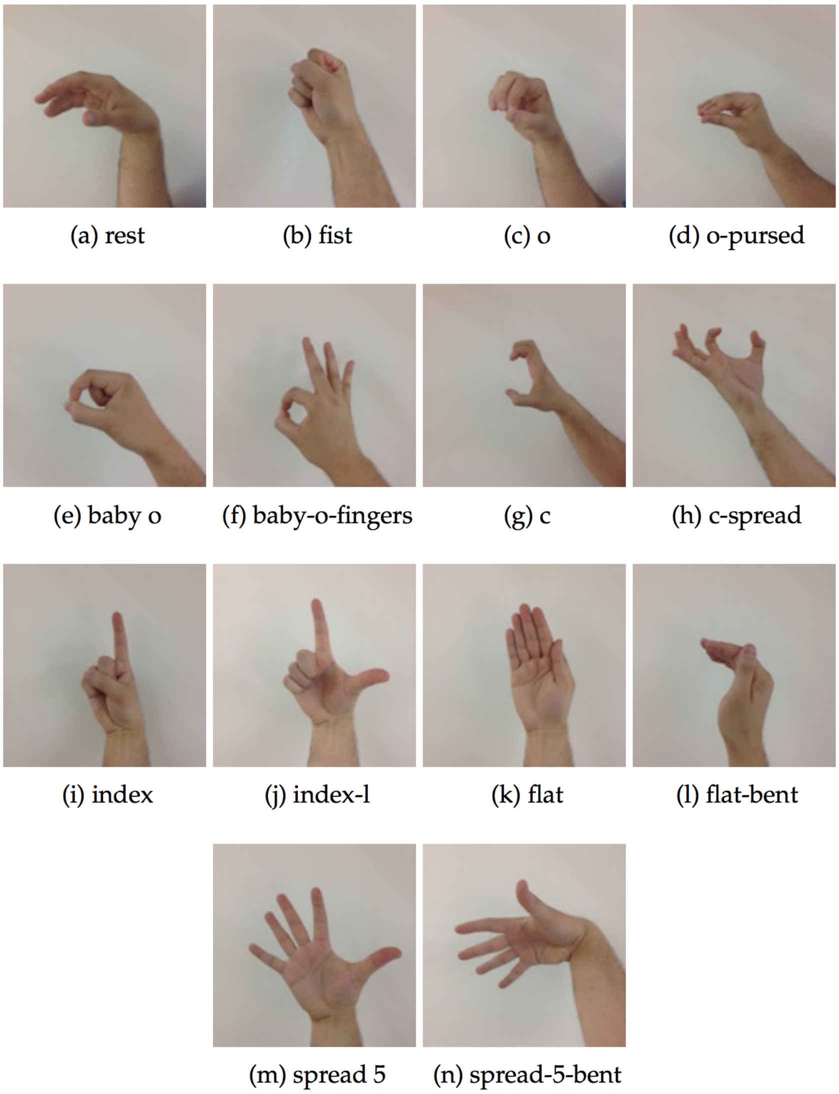
PROPOSED METHOD

In this chapter, a new algorithm for gesture recognition is presented. All steps (except feature reduction) of gesture recognition process mentioned in the background chapter of this thesis are performed. Before gesture recognition process is started, gesture set is decided. Then, the training and testing data of gestures in the set are collected. Collected data are preprocessed and features are extracted. After that, the number of the feature vectors is reduced. Finally, a classifier is fed with reduced set of features and resulting gesture classes are output, which completes the gesture recognition process.

4.1 Selecting Gesture Set

Sign language recognition and device controlling are the most common application area of gesture recognition. Hence, letters or expressions of different sign languages and commanding gestures are the most recognized gestures. In order to contribute to a totally new area, expressive gestures performed by the non-dominant hand of a classical music orchestra conductor are chosen for the experiment application part of this thesis [109]. These gestures are the most commonly used gestures by conductors. Fourteen gestures including a rest pose can be seen in Figure 4.1.

In order to distinguish gestures clearly, gesture set is mostly created by attentively choosing specific gestures activating different muscle groups if EMG devices are used for data collection process. However, for some cases, sets can consist of already defined gestures some of which may have similarities. For example in [35] and [37], some of the Brazilian and American Sign Language letter representations are similar, which result in confusion on recognizing those letters. The gesture set used in this thesis also has some similar representations of some gestures. Similarity analysis will be explained in the next chapter with the results of experiments.



(a) rest (b) fist (c) o (d) o-pursed

(e) baby o (f) baby-o-fingers (g) c (h) c-spread

(i) index (j) index-l (k) flat (l) flat-bent

(m) spread 5 (n) spread-5-bent

Figure 4.1: Expressive gesture set of classical music orchestra conductor

4.2 Collecting Data Using Myo

Myo from Thalmic Labs [9] is a revolutionary device that makes EMG data collection very easy. It is a portable device that can be carried anywhere and can be used by anybody. Putting Myo armband to the thickest part of the left or right forearm is enough for starting data collection process. According to its technical specifications [110], by the help of elastic bands and sizing clips, it can be suitable for the forearms with circumference from 19cm to 34cm.

It is also stated in its technical specifications [110] that it has

- 8 Medical grade stainless steel EMG sensors which are used to recognize 5 gestures (spread fingers, fist, wave left, wave right and double tap) with Myo's build-in classifier,

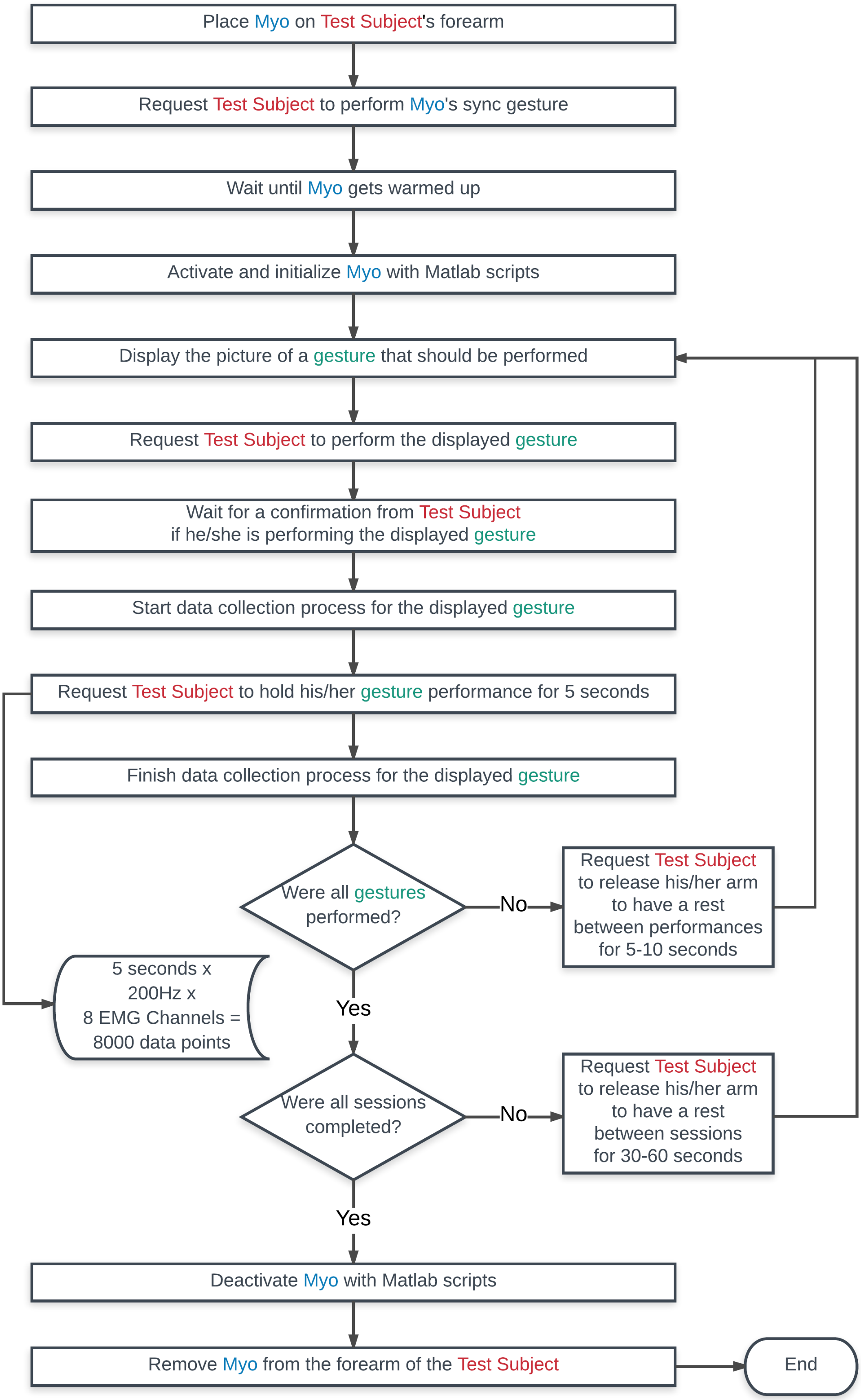

Place Myo on Test Subject's forearm
Request Test Subject to perform Myo's sync gesture
Wait until Myo gets warmed up
Activate and initialize Myo with Matlab scripts
Display the picture of a gesture that should be performed
Request Test Subject to perform the displayed gesture
Wait for a confirmation from Test Subject if he/she is performing the displayed gesture
Start data collection process for the displayed gesture
Request Test Subject to hold his/her gesture performance for 5 seconds
Finish data collection process for the displayed gesture
Were all gestures performed?
No
Request Test Subject to release his/her arm to have a rest between performances for 5-10 seconds
5 seconds x 200Hz x 8 EMG Channels = 8000 data points
Yes
Were all sessions completed?
No
Request Test Subject to release his/her arm to have a rest between sessions for 30-60 seconds
Yes
Deactivate Myo with Matlab scripts
Remove Myo from the forearm of the Test Subject
End

Figure 4.2: Flow Diagram that explains the steps of data collection process

- Highly sensitive 9-axis IMU sensors which are used to detect the orientation and position of the arm. IMU sensors contain
    - 3-axis gyroscope
    - 3-axis accelerometer
    - 3-axis magnetometer

Some of the studies [95, 98, 99, 100, 102] mentioned in the "Previous Work" chapter used those predefined gestures which are reported in real-time by Myo SDK. After Thalmic Labs announced that the updated SDK would provide raw EMG data on December 2014 [111], many researchers started developing algorithms for their own gesture recognition studies.

Matlab scripts are used to collect data by using Myo. There is no official Matlab SDK version by Thalmic Labs, however Mark Tomaszewski developed Myo SDK Matlab Mex Wrapper that used C++ bindings of official Myo SDK [112]. His wrapper is capable of streaming EMG data with the maximum frequency of 200Hz which is equal to the Myo's data acquisition frequency. His algorithm does not ask device for one set of data for each constant interval of 1/200 seconds. Instead it asks less frequently but obtains more than one set of data. It writes the obtained data to buffers, which prevents data loss.

In Figure 4.2, steps of data collection process are explained in detail.

At the end of the data collection process for each user, 560000 data points (5 seconds x 200Hz x 8 EMG channels x 14 gestures x 5 times) are collected with EMG sensors. Sample EMG signals for the fist gesture collected in a single session can be seen in Figure 4.3.

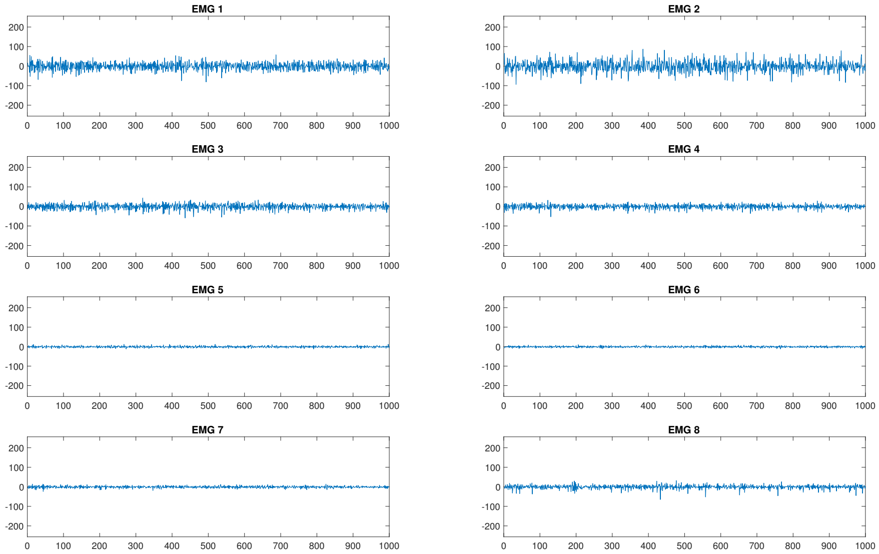


Figure 4.3: 8-channel sample EMG signals for fist gesture

4.3 Pre-processing

For the pre-processing step of the proposed gesture recognition system, Full-Wave Rectification (FWR) and Moving Averaging Window (MAW) are applied to raw EMG signals, respectively.

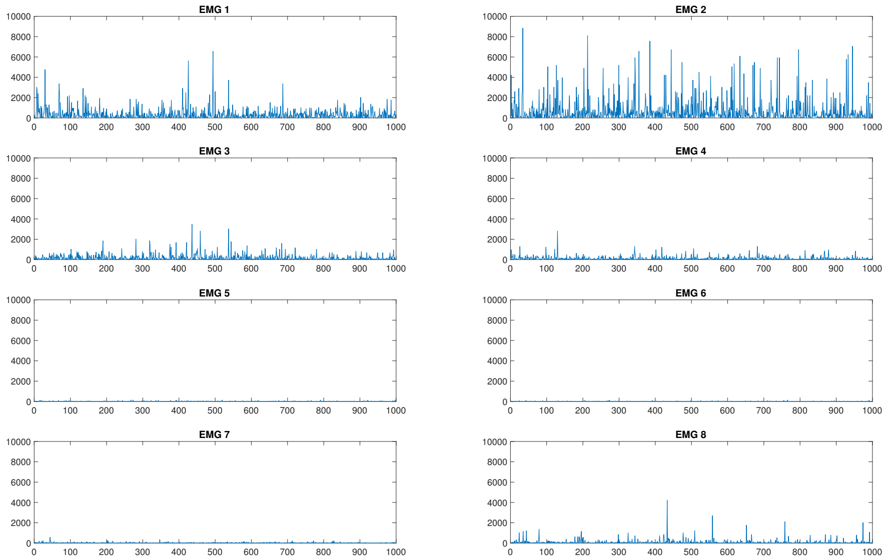


Figure 4.4: Signal plots after applying FWR to the data in Figure 4.3

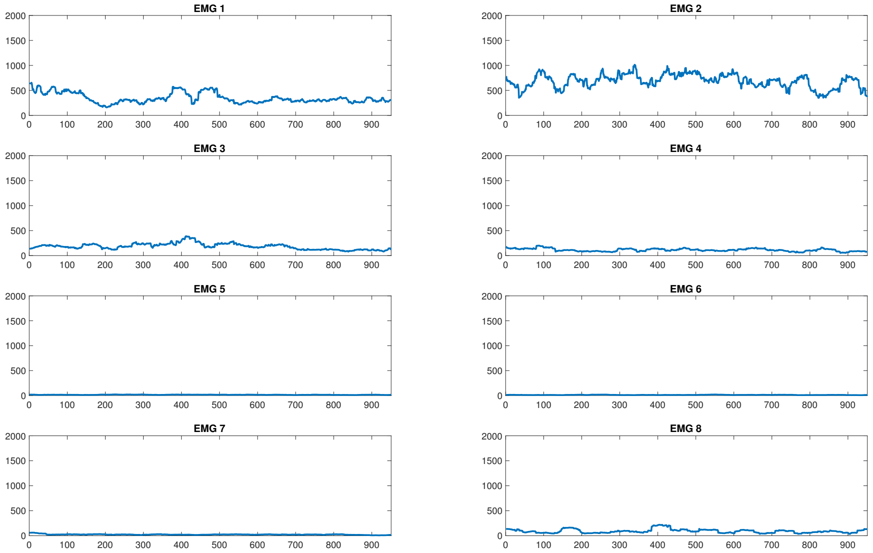


Figure 4.5: Signal plots after applying MAW to the data in Figure 4.4

The reason why FWR is applied is to obtain the energy level in the signal. High values on EMG channel can correspond to more muscle activity on the muscles contacting with that specific EMG sensor. Sample data after applying FWR is plotted in Figure 4.4.

After applying FWR to raw data, MAW is applied on pre-processed data. The main purpose of applying MAW is to smooth the signal data. Plot of the resulting signal can be seen in Figure 4.5.

Values from 20 to 100 is tested for window size and it is specified as 50 which is found as optimal. When the window size is getting closer to 20, smoothness decreases. When the window size is getting closer to 100, latency occurs while detecting gestures.

Averaging window is applied to data by sliding it by one data point for each iteration. When the window size is specified as 50, the new data with index 1 is equal to the mean of old data with index between 1 and 50. Similarly, the new data with index 2 is equal to the mean of old data with index between 2 and 51. This algorithm is executed on the data set until the last data point is included in the averaging calculation. MAW algorithm is explained in detail in Equation 4.1

$$\mathrm{MAW}_i(n) = \sum_{p=n}^{n+M-1} W(n)|X_i(p)| \tag{4.1}$$

where M represents the size of the window. An averaging window described in Equation 4.2 was used.

$$W(n) = \frac{1}{M} \tag{4.2}$$

This also reduces the size of data vector. It is calculated as in Equation 4.3.

$$\#\mathrm{VectorSize}_{\mathrm{new}} = \#\mathrm{VectorSize}_{\mathrm{old}} - M + 1 \tag{4.3}$$

After applying two different pre-processing methods to collected data, the system is ready for extracting features.

4.4 Extracting Feature Vectors

Feature vectors are the most effective element for accurate gesture recognition. Chosen features directly contribute to the success rate of the system. Thus, feature vectors should be specified very carefully.

Including as many features as possible helps obtaining high accuracy results. However, there is a trade-off between accuracy and performance. More feature vectors in number increase the size of the data for calculations, which results in performance loss.

Results of many combinations of different feature vectors were investigated. Finally, 5 feature vectors which gave the most accurate classification results with low latency were decided to be used in the classification step.

4.4.1 Moving Averaging Window

Pre-processed data is directly used as the first feature vector. This pre-processed 8-channel EMG data hold the information of the energy exerted by forearm muscles. When continuous EMG signals have been observed, it is seen that different channels are activated during the performance of different gestures. Hence, this information is regarded as valuable to select as a feature.

4.4.2 FWR of Inter-Element Differences

The second selected feature is FWR of inter-element differences. This feature is calculated as getting second powers of values obtained with the Equation 2.1. This formula is applied to raw data instead of pre-processed data because IED is meaningful when signs of values of a signal changes rapidly. Otherwise, the differences approach to 0 when the formula is applied to pre-processed, in other words smoothed, data. Figure 4.6 shows the resulting signal plots after applying IED to the raw data.

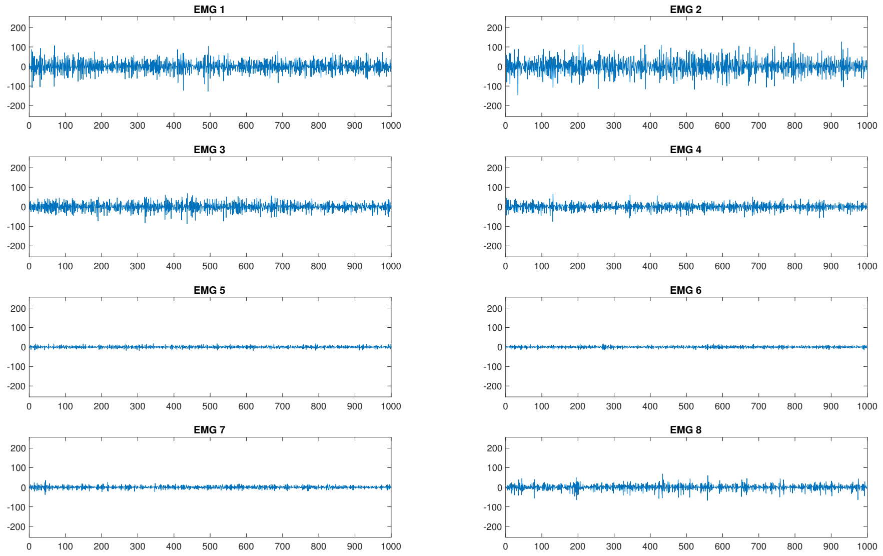


Figure 4.6: Signal plots after applying IED to the data in Figure 4.3

Squaring IED values emphasize the amount of differences. Therefore, FWR is applied to the resulting signal which can be seen in Figure Figure 4.7.

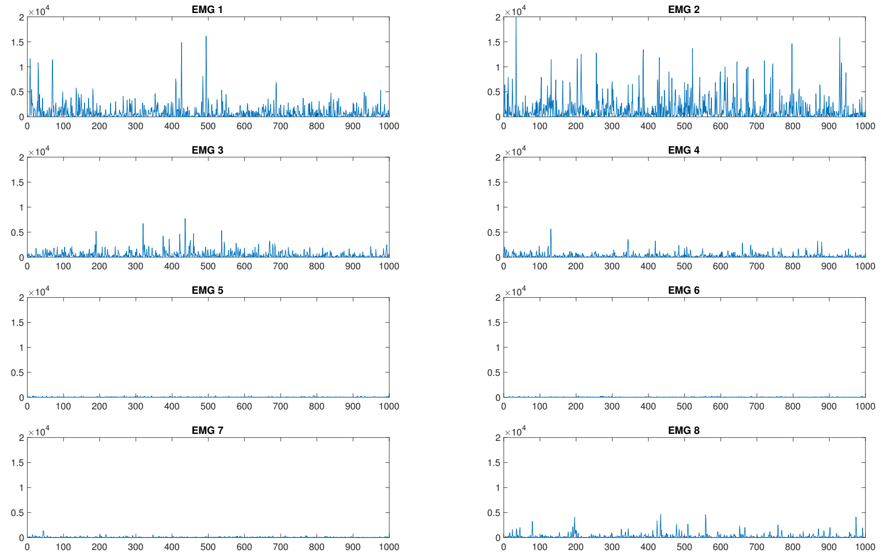


Figure 4.7: Signal plots after applying FWR to the data in Figure 4.6

MAW is also applied to the obtained data to smooth the signal. Resulting signals of after applying MAW are plotted in 4.8.

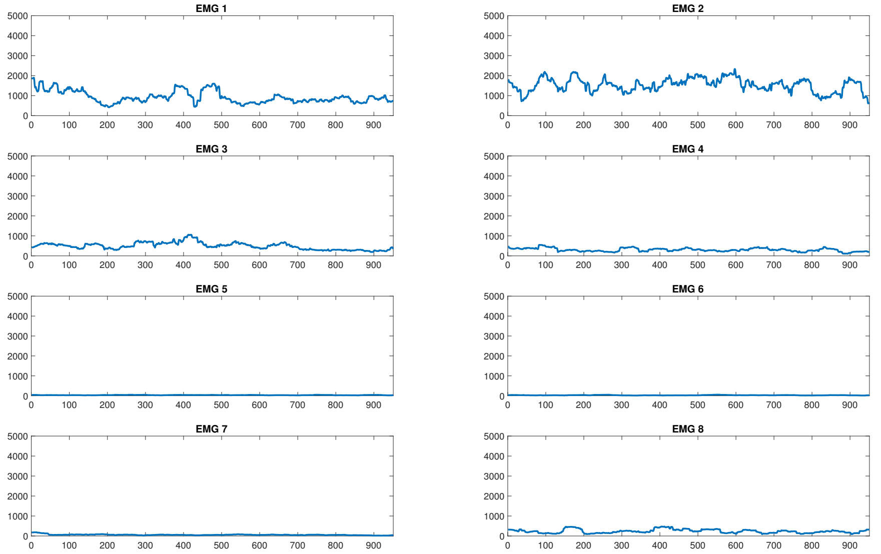


Figure 4.8: Signal plots after applying MAW to the data in Figure 4.7

4.4.3 Circular Harmonic Coefficients

The main idea to use circular harmonic coefficients as a feature vector is that the signal pattern of each gesture is unique and each signal can be represented with combinations of circular harmonic functions. Thus, it is assumed that the circular harmonic coefficients for each gesture signal are distinct enough to distinguish one gesture from the other.

Circular harmonic coefficients are calculated according to the Equation 2.10 where the basis function is revealed in Equation 2.8.

Myo on the forearm is represented in polar coordinate system as in Figure 4.9. Myo's 8 EMG channels are assumed to be placed on the polar coordinate system with $\frac{\pi}{4}$ differences.

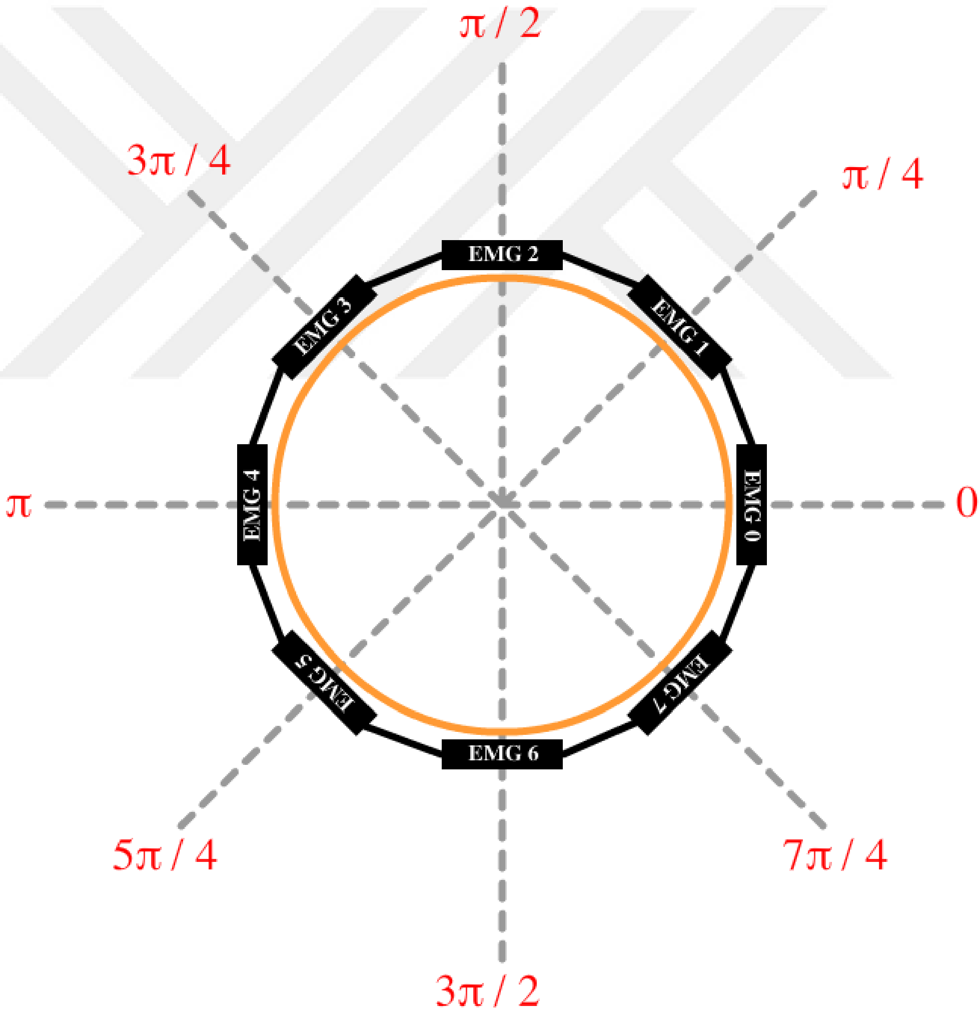


Figure 4.9: Myo representation on polar coordinate system

In order to break the dependency of θ, integral in the Equation 2.10 is converted into summation by dividing the range of 0 to 2π into 8 equal pieces with $\dfrac{\pi}{4}$ differences. By changing $f(\theta)$ with corresponding EMG values and placing the value of basis function, the equation becomes:

$$w_n = \sum_{r=0}^{r=7} \mathrm{EMG}_r \times e^{-i\times n\times(r\times\frac{\pi}{4})} \tag{4.4}$$

where n is the order of circular harmonic coefficient. 8 orders are calculated and used to extract 3 different features. Resulting coefficients are complex numbers which is in the $a + bi$ form where a and b are real numbers and i is the imaginary unit. The features extracted from circular harmonic coefficients are:

1. Real part (a)
2. Imaginary part (b)
3. Magnitude ($\sqrt{a^2 + b^2}$)

It should also be noted that pre-processed data filtered with MAW is used instead of raw EMG signals for calculations.

At the end of feature extraction step, for each of 14 gestures, 5 different feature vectors each of whose dimensions are 8×951 are obtained to use in classification step.

4.5 Reducing the Number of Feature Vectors

At the end of the feature extraction step, 40-dimension-feature set was obtained. In order to reduce the number of the dimensions, PCA was applied. The algorithm was applied to one test subject's data in order to determine the optimal number of principle components (PCs). After applying PCA to the data, a new 40-dimension-feature set containing PCs was obtained. In order to find the effective principal components, classification step which will be explained in the following section was applied. Forty different classifier was initialized with different number of principal components, *i.e.*, the first classifier used the first principal component, the second classifier used the first and the second principal components, the third classifier used the first, the second and the third principal components. One more classifier was initialized without applying PCA to the feature vectors to compare the classification results. In Figure 4.10, results of PCA-applied-classification and PCA-not-applied-classification can be compared.

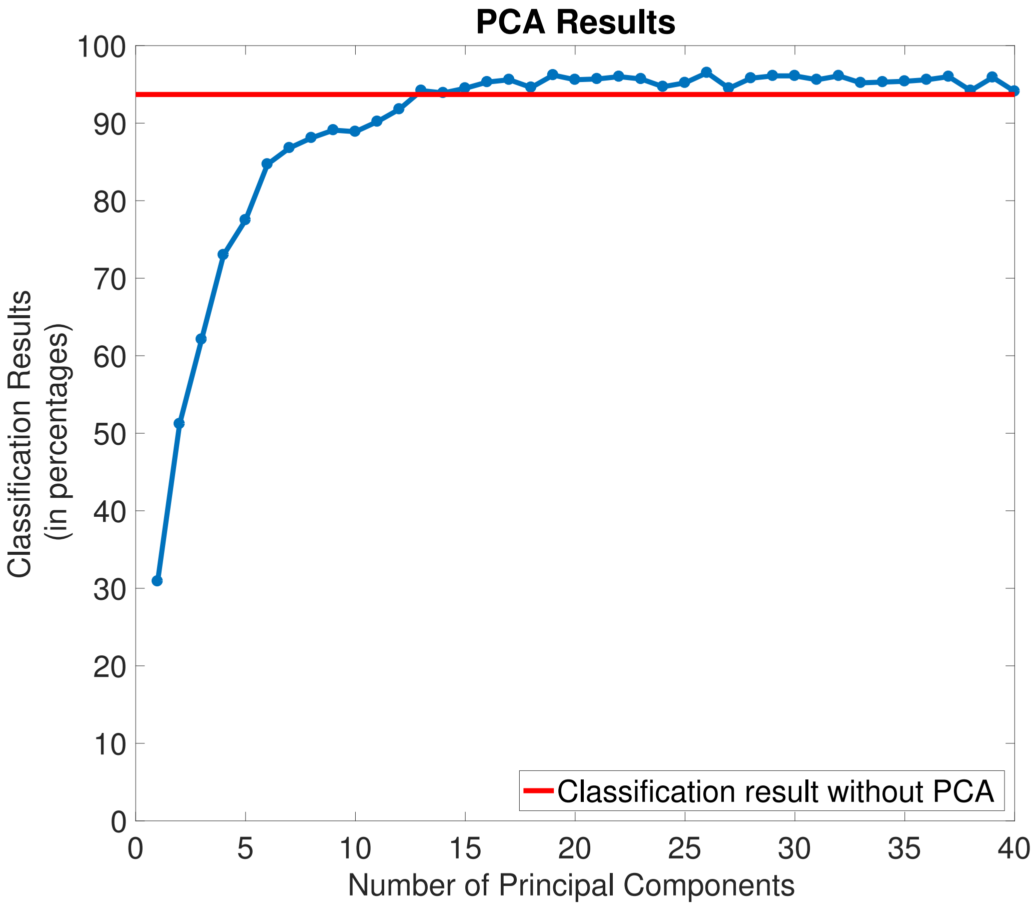


Figure 4.10: Comparison of the classification results with and without applying PCA

From Figure 4.10, it can be inferred that after the thirteenth principal component the classification results are not significantly change as the number of principal components are increased. In other words, the first thirteen principal components can be used instead of the original 40-dimension-feature set.

4.6 Performing Classification

For classification purpose, which is the final step of gesture recognition process, ANN is used. Matlab's Neural Pattern Recognition Tool [113] which creates a two-layer feed-forward network with sigmoid hidden and softmax output neurons is selected. $patternnet$ function with default training function $trainscg$ and default performance function $crossentropy$ is executed to create the network. Different hidden layer sizes from 10 (default) to 50 are tested for accuracy and performance. Twenty as the hidden layer size is found as an optimal value. Resulting ANN can be seen in Figure 4.11.

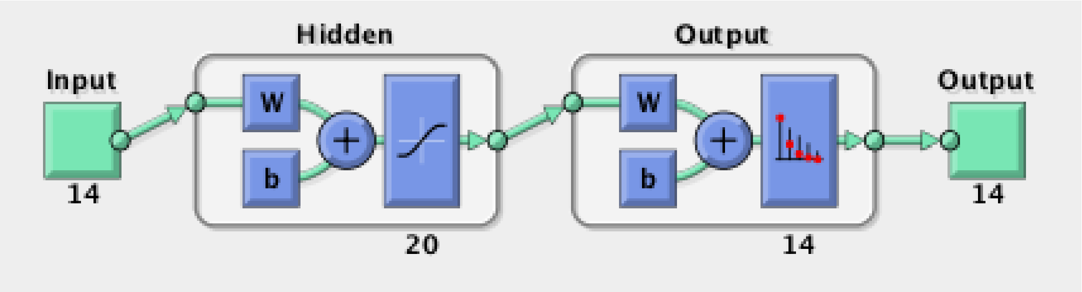


Figure 4.11: Resulting ANN with 14 principle components, 20 hidden layers and 14 outputs

For data dividing function of the network, two different functions are used:

- *dividerand* distributes the input data randomly among training, validating and testing data sets.
- *divideblock* distributes the input data with blocks among training, validating and testing data sets with respect to the following ratios:
  - *divideParam.trainRatio*
  - *divideParam.valRatio*
  - *divideParam.testRatio*

dividerand is the default data dividing function of the network. Data distribution with *dividerand* is accurate when it is used with big sample sizes with high number of outputs. For this study, number of outputs is the number of gestures to be recognized.

divideblock is used when training, validating and testing data division is important. Reasons of using *divideblock* as a data division function, together with the conducted experiments and obtained gesture recognition results are explained in detail in the following chapter.

CHAPTER 5

EXPERIMENTS AND RESULTS

This chapter presents conducted experiments and results of those experiments for the proposed generic hand and finger gesture recognition system. Three different experiments are conducted and details of each experiment are explained in the following sections.

To be used in tables and figures; gesture name and abbreviation mappings are given in 5.1.

Table 5.1: Gesture names and corresponding abbreviations

Gesture Name	Abbreviation
rest	R
fist	F
o	O
o-pursed	OP
baby-o	BO
baby-o-fingers	BOF
c	C
c spread	CS
index	I
index-l	IL
flat	FL
flat-bent	FLB
spread-5	S5
spread-5-bent	S5B

5.1 System Validation

The first experiment is the system validation experiment. In order to evaluate the success rate of the proposed system, person-dependent off-line experiment was conducted. In this experiment, 10 volunteer test subjects were chosen and gesture recognition process was executed for each test subject. The process started with data collection step. For 14 gestures which can be seen in Figure 4.1, 14 steps of data collection mentioned in Section 4.2 were performed.

In this experiment, test subjects were requested to perform each gesture by exerting the similar (if it is possible, the same) strength. It was proved by Abreu *et al.* in [35] that exerting different amount of strength would change the structure of the signal obtained from EMG sensors.

At the end of the data collection step, data of 14-gesture-set was collected 5 times within the same half an hour session of each test subject.

Collected data were input to the proposed system. In classification step, 2 different methods were applied. Details of these methods are explained in detail in the following subsections.

5.1.1 Using *dividerand* for ANN's data dividing function

In the first method, *dividerand* was used for ANN's data dividing function. 70% of the data was used for training, 15% of the data was used for validating and 15% of the data was used for testing the network. The main feature of this method is to randomly separate the whole data between training, validating and testing phases. In the training phase of the classification part, 7 different algorithms are compared. These algorithms are listed below:

1. ***scg***: Scaled conjugate gradient backpropagation

2. ***bfg***: BFGS quasi-Newton backpropagation

3. ***cgb***: Conjugate gradient backpropagation with Powell-Beale restarts

4. ***cgf***: Conjugate gradient backpropagation with Fletcher-Reeves updates

5. ***cgp***: Conjugate gradient backpropagation with Polak-Ribiére updates

6. ***oss***: One-step secant backpropagation

7. ***rp***: Resilient backpropagation

Classification results of each test subject with 7 different training algorithms can be compared in Table 5.2. In all calculations, PCA was applied to the feature set and the first 14 principal components were used as features.

Table 5.2: Classification results (in percentages) with 7 different training functions (PCA was applied and the first 14 PCs were used)

	scg	*bfg*	*cgb*	*cgf*	*cgp*	*oss*	*rp*
TS 1	93.9%	92.6%	94.4%	93.6%	94.1%	87.8%	92.1%
TS 2	92.9%	95.8%	95.7%	94.5%	94.5%	89.9%	92.2%
TS 3	94.7%	93.9%	93.5%	91.1%	95.1%	92.0%	90.5%
TS 4	96.2%	94.0%	93.8%	95.7%	96.0%	92.1%	94.2%
TS 5	98.2%	96.7%	98.3%	97.6%	98.1%	96.0%	96.6%
TS 6	96.6%	95.4%	96.2%	95.9%	96.4%	92.5%	94.3%
TS 7	97.1%	94.2%	96.0%	96.0%	96.8%	91.2%	93.3%
TS 8	89.1%	88.7%	87.1%	84.2%	88.8%	83.4%	85.5%
TS 9	95.4%	95.8%	95.4%	95.3%	95.3%	90.7%	93.5%
TS 10	96.4%	94.4%	95.9%	94.0%	95.0%	87.9%	91.7%
Avg	95.05%	94.15%	94.63%	93.79%	95.01%	90.35%	92.39%

Table 5.3: Training time results (in seconds) with 7 different training functions (PCA was applied and the first 14 PCs were used)

	scg	*bfg*	*cgb*	*cgf*	*cgp*	*oss*	*rp*
TS 1	129	182	298	233	352	218	116
TS 2	69	302	264	359	255	269	113
TS 3	117	204	165	145	307	307	115
TS 4	92	154	87	159	169	187	117
TS 5	107	144	169	160	220	320	75
TS 6	104	198	136	128	151	184	59
TS 7	124	138	168	189	235	181	102
TS 8	132	220	211	140	217	296	115
TS 9	79	199	176	128	148	164	108
TS 10	123	190	283	210	219	201	120
Avg	107.6	193.1	195.7	185.1	227.3	232.7	104.0

From Table 5.2 and Table 5.3, it can be seen that *scg* algorithm gives the highest classification results where *rp* algorithm gives the lowest training time results comparing to other algorithms. There are 3.6 seconds difference between *scg* and *rp* algorithms in terms of average training time, which can be regarded as insignificant. On the other hand, there are 2.66% difference between *scg* and *rp* algorithms in terms of success rate, which can be regarded as significant. As a result, *scg* algorithm was preferred with the average success rate of 96.09% because successful classification is more important than training time in the scope of this thesis.

Table 5.4: Classification results (in percentages) with 7 different training functions (PCA was applied and the first 40 PCs were used)

	scg	*bfg*	*cgb*	*cgf*	*cgp*	*oss*	*rp*
TS 1	95.6%	94.0%	95.4%	92.6%	96.0%	91.6%	94.3%
TS 2	95.5%	93.2%	91.1%	95.3%	95.3%	90.9%	92.3%
TS 3	95.6%	95.6%	96.2%	94.9%	95.1%	90.3%	91.9%
TS 4	96.2%	96.1%	96.2%	95.8%	96.4%	92.9%	95.7%
TS 5	98.8%	96.1%	98.6%	98.9%	99.0%	96.7%	98.2%
TS 6	97.9%	95.4%	97.0%	96.0%	97.3%	94.2%	96.1%
TS 7	97.4%	96.9%	97.7%	95.4%	94.6%	94.9%	94.4%
TS 8	89.4%	87.7%	90.4%	83.6%	90.9%	85.2%	86.4%
TS 9	97.7%	93.2%	95.4%	95.6%	96.2%	92.3%	93.9%
TS 10	96.8%	94.2%	95.6%	96.2%	95.4%	91.4%	93.4%
Avg	96.09%	94.24%	95.36%	94.43%	95.62%	92.04%	93.66%

Table 5.5: Training time results (in seconds) with 7 different training functions (PCA was applied and the first 40 PCs were used)

	scg	*bfg*	*cgb*	*cgf*	*cgp*	*oss*	*rp*
TS 1	137	321	303	200	425	337	136
TS 2	164	446	139	333	406	381	134
TS 3	139	609	411	279	266	260	117
TS 4	110	407	158	163	159	174	116
TS 5	145	227	211	290	370	265	126
TS 6	125	313	211	221	206	257	86
TS 7	208	442	302	473	193	358	145
TS 8	170	400	355	186	488	360	147
TS 9	253	248	169	206	280	222	95
TS 10	210	373	243	313	290	276	150
Avg	166.1	378.6	250.2	266.4	308.3	289.0	125.2

In Table 5.4 and Table 5.5, classification and training time results which were obtained by applying PCA and using the total of 40 PCs as features were shared. The comparison of using 14 and 40 PCs were given in Table 5.6.

Table 5.6: Classification and training time comparison of using 14 and 40 PCs

	scg	*bfg*	*cgb*	*cgf*	*cgp*	*oss*	*rp*
Average classification results (14 PCs)	95.05%	94.15%	94.63%	93.79%	95.01%	90.35%	92.39%
Average classification results (40 PCs)	96.09%	94.24%	95.36%	94.43%	95.62%	92.04%	93.66%
Difference (in %)	1.04%	0.09%	0.73%	0.64%	0.61%	1.69%	1.27%
Average training time results (14 PCs)	107.6	193.1	195.7	185.1	227.3	232.7	104.0
Average training time results (40 PCs)	166.1	378.6	250.2	266.4	308.3	289.0	125.2
Difference (in %)	35.21%	48.99%	21.78%	30.51%	26.27%	19.48%	16.93%

According to Table 5.6, there are 1.04% decrease in the classification success rate where there are 35.21% improvement on average training time results with the selected *scg* algorithm for training.

Gesture recognition accuracy results for the first experiment with *dividerand* function used for ANN's data dividing function seem satisfactory because the system is configured person and session dependent. It is also evaluated off-line, meaning that the data collection step is performed before classification instead of being performed simultaneously.

5.1.2 Using *divideblock* for ANN's data dividing function

In the second method, *divideblock* was used for ANN's data dividing function in order to check the consistency between collected 5 data sets for each test subject. All permutations of data sets for cross testing can be seen in Table 5.7.

For each test subject, 20 ANNs were trained, validated and tested with the corresponding data sets listed in Table 5.7. Minimum, maximum and average results of each test subject are listed in Table 5.8.

Table 5.7: Data set setups for ANN with *divideblock* dividing function

Setup	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Train	1,2,3	1,2,3	1,2,4	1,2,4	1,2,5	1,2,5	1,3,4	1,3,4	1,3,5	1,3,5	1,4,5	1,4,5	2,3,4	2,3,4	2,3,5	2,3,5	2,4,5	2,4,5	3,4,5	3,4,5
Validation	4	5	3	5	3	4	2	5	2	4	2	3	1	5	1	4	1	3	1	2
Test	5	4	5	3	4	3	5	2	4	2	3	2	5	1	4	1	3	1	2	1

Table 5.8: Minimum, maximum, and average results for cross testing

	Min	Max	Avg
TS 1	20.75%	74.40%	61.87%
TS 2	12.58%	72.82%	57.62%
TS 3	40.12%	74.21%	63.19%
TS 4	72.26%	85.48%	78.35%
TS 5	60.46%	85.74%	77.96%
TS 6	61.38%	84.28%	75.69%
TS 7	63.58%	77.61%	71.56%
TS 8	40.20%	70.23%	57.25%
TS 9	70.29%	87.08%	79.87%
TS 10	12.59%	74.80%	63.14%
Avg	45.42%	78.66%	68.65%

Average of the average cross testing results of 10 test subjects is 68.65%.

5.2 Basketball Referee Gestures

The second experiment was conducted by using a dataset consisting of basketball referee gestures. Portions of the research in this thesis use the MyoBR signal dataset collected by Multimedia Information System Laboratory (MISLab), Department of Computer Science and Information Engineering, National Cheng Kung University.

Yeh *et al.* organized the data which they collected using Myo for their research [101]. They decided to release it to advance the progress of HCI technology.

In their study, they collected 9 dynamic gesture data and 5 stationary gesture data from 11 test subjects. For each test subject, they collected 3-channel-ACC and 8-channel-EMG data in total of 10 different sessions. Number of collected data for each gesture is ≈155 for ACC and ≈590 for EMG.

A small part of the whole data was used in order to use with the proposed algorithm in this thesis. 10-session-EMG data of 5 stationary gestures of the first test subject were organized to match with the input layout of the proposed algorithm. For each gesture data, first 580 8-channel-EMG data points were considered to make all inputs have the same number of data. Then, the data of all sessions were concatenated. The proposed algorithm was input with the 100% of resulting dataset and success results can be seen in the first column of Table 5.9.

Table 5.9: Classification results of the proposed algorithm which was input with the basketball referee gesture data

	100% of data	50% of data (first 25% and last 25% were excluded)	75% of data (only first 25% were excluded)
TS 1	68.5%	80.1%	89.4%
TS 2	66.7%	79.2%	90.5%
TS 3	74.8%	87.5%	91.9%
TS 4	69.2%	86.8%	88.0%
TS 5	71.5%	90.0%	93.5%
TS 6	76.7%	82.1%	93.1%
TS 7	65.0%	73.8%	82.9%
TS 8	76.9%	87.9%	90.1%
TS 9	69.2%	81.1%	87.8%
TS 10	65.9%	82.3%	86.4%
TS 11	64.1%	72.1%	87.7%
Avg	69.8%	82.1%	89.2%

Relatively low success rates comparing to the previous experiment were obtained. The main reason for this was found to be the differences in the data collection steps. In the scope of this thesis, data collection was started after the test subject performs the gesture. However, in this study [101], firstly data collection was started, then the test subject performs gesture. As a result, dataset for each gesture contains the data of the transition from hand rest position and the related gesture.

Therefore, the same procedure was repeated two times with the dataset,

- whose first 25% and the last 25% were ignored (50% of the whole data).
- whose first 25% were ignored (75% of the whole data).

Results of these experiments can be seen in the second and third columns of Table 5.9 respectively.

Although the proposed gesture recognition system was not designed as subject independent, in order to compare the results with the results obtained in the original study [101], another sub-experiment was conducted. Firstly, Leave-One-Participant-Out-Cross-Validation (LOPOCV) was applied to the whole data. 80% of data of 10 test subjects were used for training, 20% of the data of the same 10 test subjects were used for validating. The data of remaining 11th test subject were used for testing. Classification results can be seen in Table 5.10. Average classification rate of 41.5% was achieved, which is not surprising for person-dependent designed gesture recognition system.

Table 5.10: Classification results of LOPOCV

Out Participant	Percentage of correctly classified inputs
TS1	43.7%
TS2	40.1%
TS3	41.9%
TS4	41.2%
TS5	41.1%
TS6	41.5%
TS7	46.9%
TS8	38.6%
TS9	41.1%
TS10	39.8%
TS11	40.6%
Avg	41.5%

Second sub-experiment was conducted with the whole data of 11 test subjects randomly distributed to ANN. Properties of this ANN was the same with the one used in Section 5.1.1. *scg* algorithm was used for training the network. 52.1% classification accuracy was obtained. In the original study [101], 54.3% success rate was obtained.

5.3 Classical Music Orchestra Conductor Gestures

The last experiment was conducted in a classical music concert practice. Data collection process was performed naturally.

The classical music orchestra conductor was requested to wear the data collection device, Myo, on his forearm of the non-dominant hand which he uses to perform expressive gestures during his concert performances. Then he was requested to perform the concert practice as usual. Neither he was informed about any gestures in the scope of this thesis, nor he was guided about how to perform gestures. All of his gesture performance were his natural characteristic movements.

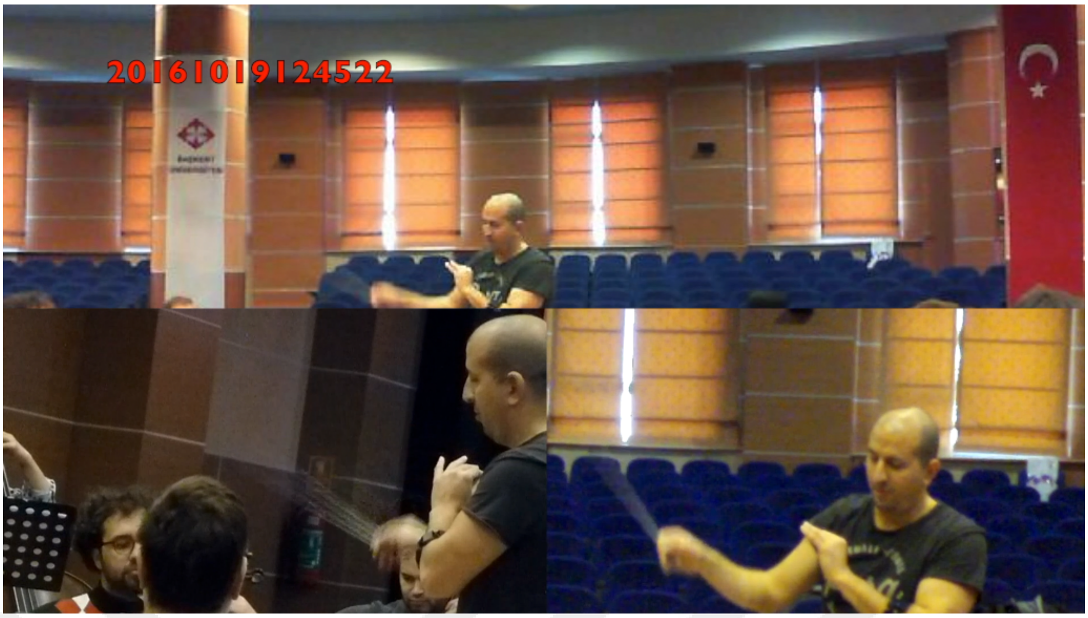


Figure 5.1: A snapshot from the concert training captured with 3 different cameras

Not all but the important parts of the concert practice were recorded with three different cameras and an advanced sound recording microphone. One of the cameras was a webcam of a computer where data collection algorithm was executed. Vision and the EMG data were collected with time stamps and they were mapped to each other while synchronizing the data. Other cameras were used to capture the gesture performances from different angles of sight. An advance sound recording microphone was used to record high quality sound which was synchronized with vision data. At the end, 67-minute-video clip was created with the combination of sound and vision data by using iMovie application [114]. A snapshot from the resulting video can be seen in Figure 5.1

After the synchronization of vision, sound and data; gesture performances were manually labeled by the supervisor by watching every second of the resulting video. Due to the fact that the data collection process was natural which means the conductor's hand movements were spontaneous, most of the gesture performances were discarded. Only the obtained data while the conductor was holding his hand obviously performing one of the gesture for at least 2 seconds were added to the data set.

Not all 14 gestures in the gesture set were performed by the conductor. This was expected because it was and mentioned in [109] that the gesture set was the combination of expressive gestures by different conductors. The number of labeled gestures performed by the conductor was 5. These gestures were similar to o-pursed, index, spread-5-bent, c-spread and baby-o-fingers. Hand rest position was also included as 6th gesture.

EMG data of labeled gesture performances are extracted from the vision-data mapping operation performed earlier. Timestamps of the collected EMG data and vision data were matched. Matching process was performed manually by the supervisor therefore numbers of samples of each gesture used in classification step varied. The gesture recognition system was input with 1992 samples of R, 1581 samples of OP, 986 samples of I, 1116 samples of S5B, 1029 samples of CS and 2032 samples of BOF.

In the classification step, ANN with $dividerand$ as data dividing function was used. 70% of the samples were used for training, 15% of samples were used for validating and remaining 15% of samples were used for testing purposes. At the end of the classification step, 96.9% accuracy was achieved for 5 gestures and the hand rest position. Confusion matrix of the classification results can be seen in Table 5.11.

Table 5.11: Classification results of the proposed algorithm which was input with the conductor training data

Gesture	Number of inputs	Number of correctly classified inputs	Number of incorrectly classified inputs	Percentage of correctly classified inputs
R	1943	1943	0	100.0%
OP	1532	1491	41	97.3%
I	937	917	20	97.9%
S5B	1067	948	119	88.8%
CS	980	969	11	98.9%
BOF	1983	1912	71	96.4%
ALL	8442	8180	262	96.9%

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter presents the conclusion of the thesis. General opinions on the topic, experiments, results and contributions are discussed in this chapter. Furthermore, potential additions for future work are offered.

6.1 Discussions

A generic gesture recognition system was developed in the scope of this thesis. However the developed system has also some shortcomings. These shortcomings and possible improvements can be summarized as follows:

- The system was developed as session dependent that means the users of the system should not take off MYO from their forearms during the whole data collection process. The reason is that the EMG sensors on MYO cannot be located on specific muscles directly because of its portability feature.

  When MYO is first worn on the forearm, its own software (MYO Connect) performs an automatic calibration process for MYO's specific position on the forearm. This calibration process is called "warm-up". During that process, MYO is forming a strong electrical connection with the muscles in its current user's forearm [115]. When this connection gets broken by taking off MYO from user's forearm, it is almost impossible to set up the same connection with the same parts of muscles. This leads to obtain different sensor values for the same performances.

  In order to break session dependency, an algorithm that loops through the sensor channels was developed. The idea came from the experiment conducted during the tests of algorithm. In that experiment, first, Myo was placed on the forearm normally and data was collected. After the first data collection session, Myo was removed from the forearm, it was rotated 45°CW and it was tried to be placed again carefully so that the previous sensor prints on the forearm could match with shifted sensors. Data collection in 8 sessions with different rotation values (0°, 45°, 90°, 135°, 180°, 225°, 270°and 315°) was performed. Before executing the proposed gesture recognition algorithm, EMG channels of collected data were shifted by 0, 1, 2, 3, 4, 5, 6, 7 respectively. Table 6.1 clearly shows the resulting EMG channel sequences for corresponding Myo rotation.

Table 6.1: EMG channel setup of Myo rotation experiment

Session	Myo rotation on forearm	EMG channel shift amount	EMG channel sequence
1	0°	0	1, 2, 3, 4, 5, 6, 7, 8
2	45°	1	2, 3, 4, 5, 6, 7, 8, 1
3	90°	2	3, 4, 5, 6, 7, 8, 1, 2
4	135°	3	4, 5, 6, 7, 8, 1, 2, 3
5	180°	4	5, 6, 7, 8, 1, 2, 3, 4
6	225°	5	6, 7, 8, 1, 2, 3, 4, 5
7	270°	6	7, 8, 1, 2, 3, 4, 5, 6
8	315°	7	8, 1, 2, 3, 4, 5, 6, 7

Then, the proposed algorithm was executed with the aforementioned EMG channel setup. Gesture recognition results can be shown in Table 6.2.

Table 6.2: Classification results of the proposed algorithm which was input with the rotation experiment data

Gesture	Number of inputs	Number of correctly classified inputs	Number of incorrectly classified inputs	Percentage of correctly classified inputs
R	7608	7358	250	96.7%
F	7608	7589	19	99.8%
O	7608	6741	867	88.6%
OP	7608	7035	573	92.5%
BO	7608	7275	333	95.6%
BOF	7608	6712	896	88.2%
C	7608	6842	766	89.9%
CS	7608	7485	123	98.4%
I	7608	7488	120	98.4%
IL	7608	7559	49	99.4%
FL	7608	7434	174	97.7%
FLB	7608	7128	480	93.7%
S5	7608	7440	168	97.8%
S5B	7608	7608	0	100.0%
ALL	106512	101694	4818	95.5%

After obtaining the success rate of 95.5%, a real-time system was developed that detected and classified the performed gesture. Firstly, an

ANN was trained with the collected data in 5 sessions. Secondly, a calibration system was developed in order to detect the rotation of the Myo on the forearm. When the test subject wore Myo, he/she was first requested to perform a specific calibration gesture. This gesture was similar to MYO's predefined "wave right" gesture which was observed to activate 2 sensors. Then the algorithm tried to find the rotation of MYO on the forearm by checking which 2 sensors had activation. After the rotation value of Myo was decided, EMG channels were shifted according to Table 6.1. Results of the real-time experiment are shared in Table 6.3.

Table 6.3: Classification results of the real-time rotation detection algorithm

Gesture	Number of trials	Number of correctly classified trials	Number of incorrectly classified trials	Percentage of correctly classified trials
R	10	10	0	100.0%
F	10	8	2	80.0%
O	10	4	6	40.0%
OP	10	3	7	30.0%
BO	10	7	3	70.0%
BOF	10	4	6	40.0%
C	10	5	5	50.0%
CS	10	3	7	30.0%
I	10	2	8	20.0%
IL	10	5	5	50.5%
FL	10	5	5	50.5%
FLB	10	3	7	30.0%
S5	10	8	2	80.0%
S5B	10	7	3	70.0%
ALL	140	74	66	52.8%

Relatively low results were obtained with respect to the results obtained by rotating Myo on the forearm (Table 6.2). When Myo is placed on the forearm randomly, it is assumed that it belongs to one of the eight rotation degree listed in Table 6.1. However, placing EMG sensors of Myo on forearm in the perfect order is not possible. This may be the reason for the low results.

As a result, in order to avoid being session dependent, using advanced EMG devices sensors of whom can be located on the same positions for each session can be recommended.

- The system was developed as user dependent that means it requires a calibration process for each user before they use the system. Not only this is related to previously mentioned MYO's sensor location problem, but also the muscular and skeletal structure of every arm is unique.

Time domain features as well as frequency domain features were used to separate gestures from each other whereas the chosen features were not enough to give accurate results for different users. The whole data collected from 10 test subjects during the system validation experiment mentioned in Section 5.1 were inputted to the developed system and 31,6% success rate is achieved, which proved that the system failed for being user independent.

Detecting each user's forearm muscle locations and using advanced EMG devices on those specific locations may provide better time domain feature values but this will not be enough to solve this issue. Hence, more useful frequency domain features may be determined to overcome user dependency problems.

- The system works well when all calculations are performed off-line however real-time accuracy should be improved because most of the applications needs real-time gesture recognition. There are averaging filters used in the system that makes gesture recognition more accurate but with more delay. Different types of averaging filters such as Gaussian Filter may be used to obtain higher success rates on real-time results.

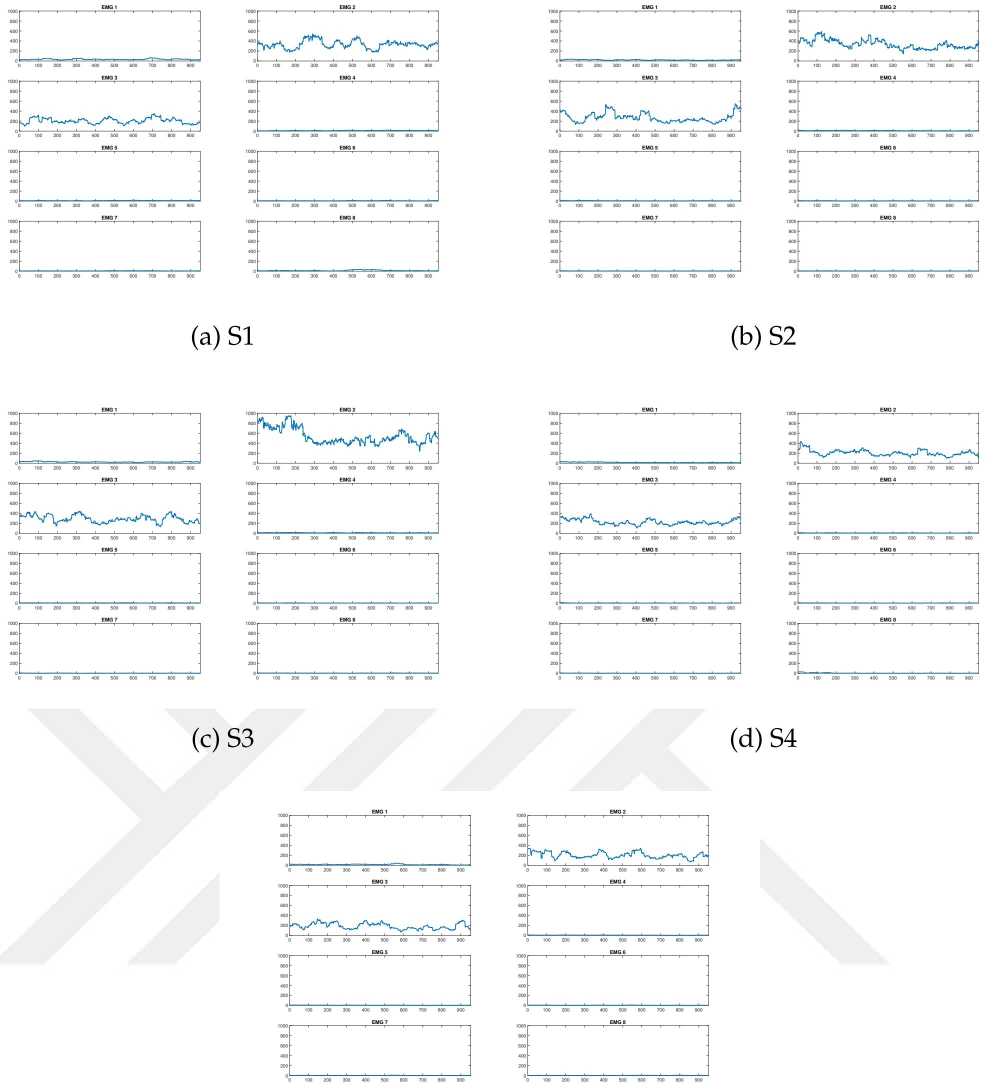
- The gesture set used in this thesis may not be appropriate for gesture recognition by using MYO device. It is aforementioned many times that MYO is a restricted device in terms of sensor positioning. Thus, gesture set should be formed by choosing distinctive gestures.

Table 6.4: Comparison of the minimum, maximum, and average results for cross testing of 13+1 gesture data set and 5+1 gesture data set

	13+1 gestures			5+1 gestures		
	Min	Max	Avg	Min	Max	Avg
TS 1	20.75%	74.40%	61.87%	66.54%	91.87%	85.83%
TS 2	12.58%	72.82%	57.62%	71.58%	89.71%	81.20%
TS 3	40.12%	74.21%	63.19%	56.51%	86.41%	77.79%
TS 4	72.26%	85.48%	78.35%	78.75%	92.06%	84.68%
TS 5	60.46%	85.74%	77.96%	71.10%	90.82%	83.46%
TS 6	61.38%	84.28%	75.69%	74.90%	91.22%	85.96%
TS 7	63.58%	77.61%	71.56%	77.87%	91.07%	85.72%
TS 8	40.20%	70.23%	57.25%	50.54%	77.01%	68.95%
TS 9	70.29%	87.08%	79.87%	86.67%	93.39%	90.56%
TS 10	12.59%	74.80%	63.14%	71.46%	90.50%	82.17%

However in this case, the gesture set consists of real life gestures that classical music orchestra conductors uses during concert performances.

Moreover, number of the gestures is another factor that reduces precision of the gesture recognition. This is the main reason why MYO developers limited the number of recognized gestures with only 5 gestures.



(a) S1

(b) S2

(c) S3

(d) S4

(e) S5

Figure 6.1: Signal plots of pre-processed fist gesture data of TS8 during 5 consecutive data collection sessions

Table 6.5: Mean values of each EMG channel shown in Figure 6.1

	EMG1	EMG2	EMG3	EMG4	EMG5	EMG6	EMG7	EMG8
S1	34	332	205	11	9	7	6	13
S2	21	335	270	9	6	5	4	4
S3	26	540	291	11	7	5	4	5
S4	16	207	226	7	5	5	4	6
S5	18	206	187	6	5	4	3	3

In order to prove the increase in the accuracy with less number of gestures, a sub-experiment was conducted. Data of 5 gestures and a hand rest position used by the conductor in the experiment explained in Section 5.3 were picked from the same dataset collected from 10 test subjects as explained in Section 5.1. The same cross testing procedure was executed with these set of data. The huge increase in the success percentages of minimum, maximum and average results can be seen in Table 6.4.

- Consecutive data collection sessions may cause muscle fatigue for some test subjects. Test subjects are requested to perform 14 gestures in 5 sessions. Each gesture performance takes 5 seconds. Figure 6.1 shows the plots of pre-processed fist gesture data for 5 consecutive sessions of TS8.

  From Table 6.5, it can be inferred that the fist gesture cause more activation on the forearm muscles coupled with EMG2 and EMG3 sensors for TS8. In third session, there is a significant increase in EMG2 sensor, which is followed by a rapid decrease in forth session. Similar trend is seen for EMG3 sensor yet slightly comparing to EMG2 sensor. The reason can be that TS8 may get tired by performing gestures through the consecutive sessions.

Table 6.6: Classification results with 1-second-gesture performances instead of 5-second-gesture performances

Gesture	Number of inputs	Number of correctly classified inputs	Number of incorrectly classified inputs	Percentage of correctly classified inputs
R	755	755	0	100.0%
F	755	753	2	99.7%
O	755	744	11	98.5%
OP	755	740	15	98.0%
BO	755	755	0	100.0%
BOF	755	743	12	98.4%
C	755	753	2	99.7%
CS	755	755	0	100.0%
I	755	755	0	100.0%
IL	755	750	5	99.3%
FL	755	754	1	99.9%
FLB	755	738	17	97.7%
S5	755	755	0	100.0%
S5B	755	755	0	100.0%
ALL	10570	10505	65	99.4%

One approach may be decreasing the gesture performance time. Test subjects were requested to hold their hands and fingers stable during the gesture performance which lasts 5 seconds for each gesture. Classification accuracy results of decreasing gesture performance time from 5 seconds to 1 second can be seen in Table 6.6. The reason for obtaining almost perfect results can be the decreased total session time. However, when real-time application areas of the proposed system is considered, especially classical music orchestra conducting, the gesture performances will be continuous. Moreover, the gesture performer hand and fingers will be active during the concert. Therefore, the proposed system should be trained and tested with the data collected during longer sessions.

6.2 Contributions

Using circular harmonic coefficients and different variations of those coefficients as frequency domain feature sets is a novel approach, which increases the gesture recognition accuracy 1.92% in average in comparison to using only time domain feature vectors. Detailed comparison of the classification results with different feature vector sets can be seen in Table 6.7

Table 6.7: Classification results with different feature sets

			Circular Harmonic Coefficients				
	MAW	FWR of IED	Real	Imag.	Magn.	MAW + FWR of IED	All
TS 1	88.5%	74.8%	65.2%	70.9%	86.6%	90.6%	92.3%
TS 2	93.3%	86.6%	85.8%	79.3%	91.6%	90.3%	94.3%
TS 3	92.0%	79.3%	80.5%	76.9%	92.9%	95.0%	95.0%
TS 4	92.8%	82.9%	76.9%	71.3%	89.0%	94.7%	96.1%
TS 5	97.3%	93.5%	91.1%	85.2%	96.3%	98.0%	98.7%
TS 6	95.5%	86.1%	79.2%	82.9%	91.8%	95.4%	97.3%
TS 7	95.5%	89.1%	85.4%	81.4%	93.9%	95.9%	97.8%
TS 8	82.8%	77.2%	75.3%	64.9%	81.0%	85.0%	88.3%
TS 9	94.8%	81.3%	75.9%	83.4%	93.2%	94.3%	96.1%
TS 10	93.0%	80.8%	78.2%	76.5%	92.4%	93.3%	95.8%
Avg	92.55%	83.16%	79.35%	77.27%	90.87%	93.25%	95.17%

Using expressive gestures of a classical music orchestra conductor as a gesture set in the gesture recognition system is unique. Instead of choosing the gestures which can easily be distinguishable, using a natural set of gestures is more beneficial for HCI studies because instinctive interactions always have advantages on artificial interactions.

Cross testing of the collected data is a new approach for detecting the consistency of each gesture performance subsession. Especially when using EMG devices for data collection process, the exerted strength during the performance of the same gestures in different subsessions may vary intentionally or unintentionally. These variations may cause inconsistently different data values for the same gesture, which results in low success percentages. With cross testing, problematic subsessions can be detected and omitted from the final data set.

6.3 Future Work

With the advances in the AR and VR technologies, HCI is getting more and more important. Due to the fact that gesture recognition is an inseparable part of HCI, it is inevitable that more studies will be conducted on this area.

On top of the proposed system, IMU data can be taken into consideration in order to obtain different feature vectors. Besides, IMU data can be used to detect arm movements which can be combined with the currently recognized gestures. It was mentioned that the expressive gestures are performed with non-dominant hand. By using second MYO on the forearm of the dominant hand, beat of the music can be controlled. In this way, this study can be used as a base of a virtual orchestra system.

REFERENCES

[1] Marleen B Schippers, Valeria Gazzola, Rainer Goebel, and Christian Keysers. Playing charades in the fmri: are mirror and/or mentalizing areas involved in gestural communication? *PLoS One*, 4(8):e6801, 2009.

[2] Paul Ekman, Wallace V Friesen, Maureen O'Sullivan, Anthony Chan, Irene Diacoyanni-Tarlatzis, Karl Heider, Rainer Krause, William Ayhan LeCompte, Tom Pitcairn, Pio E Ricci-Bitti, et al. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, 53(4):712, 1987.

[3] Ray L Birdwhistell. Background to kinesics. *ETC: A review of general semantics*, pages 10–18, 1955.

[4] David B Givens. The nonverbal dictionary of gestures, signs and body language cues: Adam apple-jam to zygomatic. 2002.

[5] Philip Kindred Dick, Scott Frank, and Jon Cohen. Minority Report, 2002.

[6] Microsoft Corporation. Kinect for Windows SDK from Microsoft Research. `http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/default.aspx`. accessed: 09/2016.

[7] CyberGlove Systems LLC. CyberGlove. `http://www.cyberglovesystems.com/`. accessed: 09/2016.

[8] Inc Leap Motion. Leap Motion. `https://www.leapmotion.com/`. accessed: 12/2016.

[9] Thalmic Labs. Discover MYO. `https://www.myo.com/`. accessed: 09/2016.

[10] Arduino. What is Arduino? `https://www.arduino.cc/`. accessed: 09/2016.

[11] plux | wireless biosignals. BITalino. `http://www.bitalino.com/`. accessed: 12/2016.

[12] Mithileysh Sathiyanarayanan, Tobias Mulling, and Bushra Nazir. Controlling a robot using a wearable device (myo). In *International Journal of Engineering Development and Research*, volume 3. IJEDR, 2015.

[13] Néstor Mauricio Caro Sánchez. Gesture classification based on electromyography. 2016.

[14] Olga Katzenelson and Solange Karsenty. A sign-to-speech glove. In *workshop IUI2014 on Interacting with Smart Objects, Germany*, 2014.

[15] James Hallam. Haptic mirror therapy glove: aiding the treatment of a paretic limb after a stroke. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 459–464. ACM, 2015.

[16] Ali Erol, George Bebis, Mircea Nicolescu, Richard D Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1):52–73, 2007.

[17] Zhou Ren, Jingjing Meng, and Junsong Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1–5. IEEE, 2011.

[18] Manar Maraqa and Raed Abu-Zaiter. Recognition of arabic sign language (arsl) using recurrent neural networks. In *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the*, pages 478–481. IEEE, 2008.

[19] Gert Geebelen, Steven Maesen, Tom Cuypers, and Philippe Bekaert. Real-time hand tracking with a colored glove. *3D Stereo media, Luik, Belgium*, 2010.

[20] Luigi Lamberti and Francesco Camastra. Real-time hand gesture recognition using a color glove. In *International Conference on Image Analysis and Processing*, pages 365–373. Springer, 2011.

[21] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *ACM transactions on graphics (TOG)*, volume 28, page 63. ACM, 2009.

[22] Cem Keskin, Ayse Erkan, and Lale Akarun. Real time hand tracking and 3d gesture recognition for interactive interfaces using hmm. *ICANN/ICONIPP*, 2003:26–29, 2003.

[23] C Diaz and S Payandeh. Preliminary experimental study of marker-based hand gesture recognition system. *Journal of Automation and Control Engineering Vol*, 2(3), 2014.

[24] Teach Me Anatomy. Muscles of the upper limb. `http://teachmeanatomy.info/upper-limb/muscles/`. accessed: 11/2016.

[25] Henry Gray. *Anatomy of the human body*. Lea & Febiger, 1918.

[26] Wikipedia. Anatomical terms of motion. `https://en.wikipedia.org/wiki/Anatomical_terms_of_motion`. accessed: 12/2016.

[27] Oxford English Dictionary. Oxford: Oxford university press, 1989.

[28] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[29] Yi Li. Hand gesture recognition using kinect. In *2012 IEEE International Conference on Computer Science and Automation Engineering*, pages 196–199. IEEE, 2012.

[30] Valentino Frati and Domenico Prattichizzo. Using kinect for hand tracking and rendering in wearable haptics. In *World Haptics Conference (WHC), 2011 IEEE*, pages 317–321. IEEE, 2011.

[31] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Sergio Di Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5):285–299, 2006.

[32] Prajwal Paudyal, Ayan Banerjee, and Sandeep KS Gupta. Sceptre: a pervasive, non-invasive, and programmable gesture recognition technology. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 282–293. ACM, 2016.

[33] Simone Benatti, Filippo Casamassima, Bojan Milosevic, Elisabetta Farella, Philipp Schönle, Schekeb Fateh, Thomas Burger, Qiuting Huang, and Luca Benini. A versatile embedded platform for emg acquisition and gesture recognition. *IEEE transactions on biomedical circuits and systems*, 9(5):620–630, 2015.

[34] Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *IEEE Transactions on Human-Machine Systems*, 44(2):293–299, 2014.

[35] João Gabriel Abreu, João Marcelo Teixeira, Lucas Silva Figueiredo, and Veronica Teichrieb. Evaluating sign language recognition using the myo armband. In *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, pages 64–70. IEEE, 2016.

[36] Jonghwa Kim, Stephan Mastnik, and Elisabeth André. Emg-based hand gesture recognition for realtime biosignal interfacing. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 30–39. ACM, 2008.

[37] Celal Savur. American sign language recognition system by using surface emg signal. 2015.

[38] Jianfeng Liu, Zhigeng Pan, and Xiangcheng Li. An accelerometer-based gesture recognition algorithm and its application for 3d interaction. *Comput. Sci. Inf. Syst.*, 7(1):177–188, 2010.

[39] Wikipedia. Spherical Harmonics. `https://en.wikipedia.org/wiki/Spherical_harmonics`. accessed: 12/2016.

[40] Blackpawn. Circular Harmonics. http://blackpawn.com/texts/ch/. accessed: 12/2016.

[41] Yuan-Neng Hsu and Henri H Arsenault. Optical pattern recognition using circular harmonic expansion. *Applied Optics*, 21(22):4016–4019, 1982.

[42] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):677–695, 1997.

[43] Nianjun Liu and Brian C Lovell. Hand gesture extraction by active shape models. In *Digital Image Computing: Techniques and Applications (DICTA'05)*, pages 10–10. IEEE, 2005.

[44] Rafael Muñoz-Salinas, R Medina-Carnicer, Francisco José Madrid-Cuevas, and A Carmona-Poyato. Depth silhouettes for gesture recognition. *Pattern Recognition Letters*, 29(3):319–329, 2008.

[45] Oleg Rumyantsev, Matt Merati, and Vasant Ramachandran. Hand sign recognition through palm gesture and movement. *Image Process*, 2012.

[46] Jun Cheng, Can Xie, Wei Bian, and Dacheng Tao. Feature fusion for 3d hand gesture recognition by learning a shared hidden space. *Pattern Recognition Letters*, 33(4):476–484, 2012.

[47] Jessica Lin and Dimitrios Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining*. Citeseer, 2003.

[48] Ahmad Akl, Chen Feng, and Shahrokh Valaee. A novel accelerometer-based gesture recognition system. *IEEE Transactions on Signal Processing*, 59(12):6197–6205, 2011.

[49] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[50] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[51] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[52] Wikipedia. Neuron. https://en.wikipedia.org/wiki/Neuron. accessed: 11/2016.

[53] Ericka Janet Rechy-Ramirez and Huosheng Hu. Stages for developing control systems using emg and eeg signals: A survey. *School of Computer Science and Electronic Engineering, University of Essex,*, 2011.

[54] Sofia University. Activation Function (in Neural Networks). http://www.fmi.uni-sofia.bg/fmi/statist/education/textbook/eng/glosa.html. accessed: 11/2016.

[55] Farid Parvini, Dennis McLeod, Cyrus Shahabi, Bahareh Navai, Baharak Zali, and Shahram Ghandeharizadeh. An approach to glove-based gesture recognition. In *International Conference on Human-Computer Interaction*, pages 236–245. Springer, 2009.

[56] Peter Vamplew and Anthony Adams. Recognition of sign language gestures using neural networks. *Australian Journal of Intelligent Information Processing Systems*, 5(2):94–102, 1998.

[57] Varadach Amatanon, Suwatchai Chanhang, Phornphop Naiyanetr, and Sanitta Thongpang. Sign language-thai alphabet conversion based on electromyogram (emg). In *Biomedical Engineering International Conference (BMEiCON), 2014 7th*, pages 1–4. IEEE, 2014.

[58] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

[59] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, 2001.

[60] Phil Blunsom. Hidden markov models. *Lecture notes, August*, 15:18–19, 2004.

[61] Wikipedia. Viterbi Algorithm. https://en.wikipedia.org/wiki/Viterbi_algorithm. accessed: 12/2016.

[62] Wikipedia. Viterbi algorithm. https://upload.wikimedia.org/wikipedia/commons/0/0c/An_example_of_HMM.png. accessed: 04/2017.

[63] Yun Li, Xiang Chen, Jianxun Tian, Xu Zhang, Kongqiao Wang, and Jihai Yang. Automatic recognition of sign language subwords based on portable accelerometer and emg sensors. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, page 17. ACM, 2010.

[64] Rung-Huei Liang and Ming Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 558–567. IEEE, 1998.

[65] Kenji Oka, Yoichi Sato, and Hideki Koike. Real-time fingertip tracking and gesture recognition. *IEEE Computer graphics and Applications*, 22(6):64–71, 2002.

[66] Kevin R Wheeler. Device control using gestures sensed from emg. In *Soft Computing in Industrial Applications, 2003. SMCia/03. Proceedings of the 2003 IEEE International Workshop on*, pages 21–26. IEEE, 2003.

[67] Nguyen Dang Binh, Enokida Shuichi, and Toshiaki Ejima. Real-time hand tracking and gesture recognition system. *Proc. GVIP*, pages 19–21, 2005.

[68] Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 401–406. ACM, 2009.

[69] Mario Ganzeboom. How hand gestures are recognized using a dataglove. *Human Media Interaction (HMI)*, 2009.

[70] Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. A framework for hand gesture recognition based on accelerometer and emg sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1064–1076, 2011.

[71] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[72] Santiago Ortega-Avila, Bogdana Rakova, Sajid Sadi, and Pranav Mistry. Non-invasive optical detection of hand gestures. In *Proceedings of the 6th Augmented Human International Conference*, pages 179–180. ACM, 2015.

[73] Hongyi Wen, Julian Ramos Rojas, and Anind K Dey. Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3847–3851. ACM, 2016.

[74] Jackson Taylor. Real-time translation of american sign language using wearable technology. 2016.

[75] Piyush Kumar, Jyoti Verma, and Shitala Prasad. Hand data glove: a wearable real-time device for human-computer interaction. *International Journal of Advanced Science and Technology*, 43, 2012.

[76] Francesco Camastra and Domenico De Felice. Lvq-based hand gesture recognition using a data glove. In *Neural Nets and Surroundings*, pages 159–168. Springer, 2013.

[77] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. `http://people.csail.mit.edu/rywang/hand/glove2.png`. accessed: 04/2017.

[78] Wikipedia. Kalman Filter. `https://en.wikipedia.org/wiki/Kalman_filter`. accessed: 10/2016.

[79] Yikai Fang, Kongqiao Wang, Jian Cheng, and Hanqing Lu. A real-time hand gesture recognition method. In *2007 IEEE International Conference on Multimedia and Expo*, pages 995–998. IEEE, 2007.

[80] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[81] Wikipedia. OpenNI Framework. `https://en.wikipedia.org/wiki/OpenNI#OpenNI_Framework`. accessed: 10/2016.

[82] Code Laboratories. CL NUI Platform - Kinect Driver/SDK. `https://codelaboratories.com/nui`. accessed: 10/2016.

[83] Heng Du and TszHang To. Hand gesture recognition using kinect. *Techical Report, Boston University*, 2011.

[84] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120, 2013.

[85] Wikipedia. Wii Remote. `https://en.wikipedia.org/wiki/Wii_Remote`. accessed: 10/2016.

[86] Wikipedia. PlayStation Move. `https://en.wikipedia.org/wiki/PlayStation_Move`. accessed: 10/2016.

[87] Esa Tuulari and Arto Ylisaukko-oja. Soapbox: A platform for ubiquitous computing research and applications. In *International Conference on Pervasive Computing*, pages 125–138. Springer, 2002.

[88] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[89] Wikipedia. Bessel Filter. `https://en.wikipedia.org/wiki/Bessel_filter`. accessed: 04/2017.

[90] Vasiliki E Kosmidou, Leontios J Hadjileontiadis, and Stavros M Panas. Evaluation of surface emg features for the recognition of american sign language gestures. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 6197–6200. IEEE, 2006.

[91] Celal Savur and Ferat Sahin. Real-time american sign language recognition system using surface emg signal. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 497–502. IEEE, 2015.

[92] Wikipedia. Rubik's Cube. `https://en.wikipedia.org/wiki/Rubik%27s_Cube`. accessed: 10/2016.

[93] Vasiliki E Kosmidou and Leontios J Hadjileontiadis. Sign language recognition using intrinsic-mode sample entropy on semg and accelerometer data. *IEEE transactions on biomedical engineering*, 56(12):2879–2890, 2009.

[94] Marcus Georgi, Christoph Amma, and Tanja Schultz. Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*, pages 99–108, 2015.

[95] Michael Lee and Nikhil Rao. Hand gesture data collection procedure using a myo armband for machine learning. Technical report, DTIC Document, 2015.

[96] Ali Boyali, Naohisa Hashimoto, and Osamu Matsumoto. Hand posture and gesture recognition using myo armband and spectral collaborative representation based classification. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pages 200–201. IEEE, 2015.

[97] Ali Boyali and Naohisa Hashimoto. Spectral collaborative representation based classification for hand gestures recognition on electromyography signals. *Biomedical Signal Processing and Control*, 24:11–18, 2016.

[98] Mahmoud Abduo and Matthias Galster. Myo gesture control armband for medical applications. 2015.

[99] Asilbek Ganiev, Ho-Sun Shin, and Kang-Hee Lee. Study on virtual control of a robotic arm via a myo armband for the self-manipulation of a hand amputee. *International Journal of Applied Engineering Research*, 11(2):775–782, 2016.

[100] Mithileysh Sathiyanarayanan and Sharanya Rajan. Myo armband for physiotherapy healthcare: A case study using gesture recognition application. In *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–6. IEEE, 2016.

[101] Chung-Wei Yeh, Tse-Yu Pan, and Min-Chun Hu. A sensor-based official basketball referee signals recognition system using deep belief networks. In *International Conference on Multimedia Modeling*, pages 565–575. Springer, 2017.

[102] Kristian Nymoen, Mari Romarheim Haugen, and Alexander Refsum Jensenius. Mumyo–evaluating and exploring the myo armband for musical interaction. 2015.

[103] Hideyuki Morita, Shuji Hashimoto, and Sadamu Ohteru. A computer music system that follows a human conductor. *Computer*, 24(7):44–53, 1991.

[104] Wikipedia. Glossary of musical terminology. `https://en.wikipedia.org/wiki/Glossary_of_musical_terminology`. accessed: 04/2017.

[105] Eric Lee, Ingo Grüll, Henning Kiel, and Jan Borchers. conga: A framework for adaptive conducting gesture analysis. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 260–265. IRCAM—Centre Pompidou, 2006.

[106] Teresa Marrin Nakra, Yuri Ivanov, Paris Smaragdis, and Christopher Ault. The ubs virtual maestro: an interactive conducting system. In *NIME*, pages 250–255, 2009.

[107] Takashi Baba, Mitsuyo Hashida, and Haruhiro Katayose. " virtualphilharmony": A conducting system with heuristics of conducting an orchestra. In *NIME*, volume 2010, pages 263–270, 2010.

[108] Martin Ritter and Alyssa Aska. Leap motion as expressive gestural interface. In *ICMC*, 2014.

[109] PB Braem and T Bräm. A pilot study of the expressive gestures used by classical orchestra conductors. *Journal of the Conductor's Guild*, 22(1-2):14–29, 2001.

[110] Thalmic Labs. MYO Tech Specs. `https://www.myo.com/techspecs`. accessed: 12/2016.

[111] Myo Blog. Raw and Uncut Drops Today. `http://developerblog.myo.com/raw-uncut-drops-today/`. accessed: 12/2016.

[112] Mark Tomaszewski. Myo SDK MATLAB MEX Wrapper. `https://www.mathworks.com/matlabcentral/fileexchange/55817-myo-sdk-matlab-mex-wrapper`. accessed: 12/2016.

[113] MathWorks. Classify Patterns with a Neural Network. `https://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html`. accessed: 01/2017.

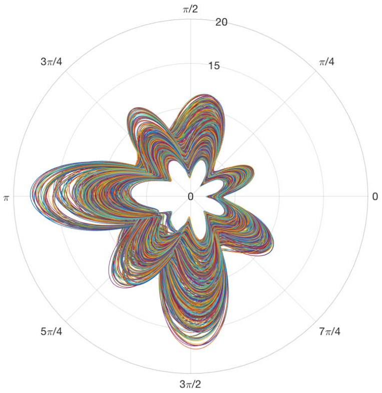
[114] Apple. iMovie. `http://www.apple.com/lae/imovie/`. accessed: 02/2017.

[115] Myo Support. Warm up while wearing your Myo armband. `https://support.getmyo.com/hc/en-us/articles/203910089-Warm-up-while-wearing-your-Myo-armband`. accessed: 04/2017.

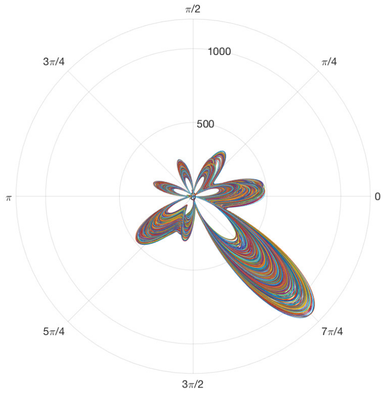
Appendix A

CIRCULAR HARMONIC FUNCTION REPRESENTATIONS OF 14 GESTURES

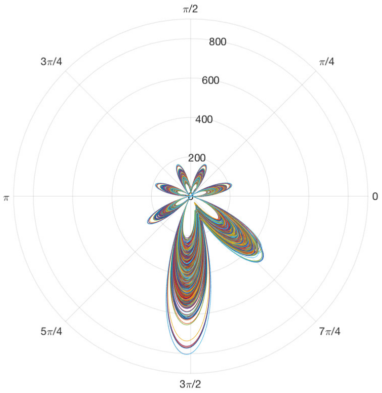
free



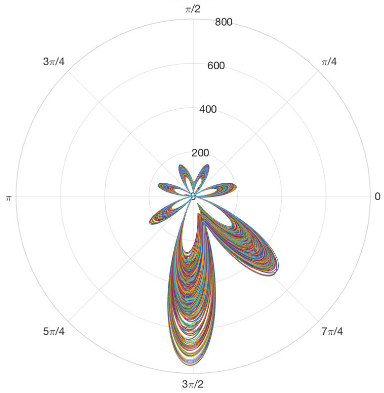
fist

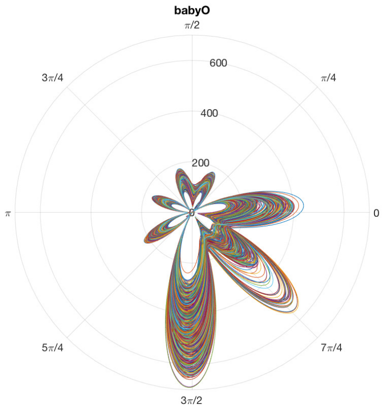


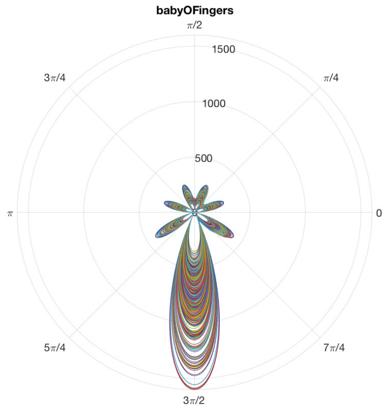
o

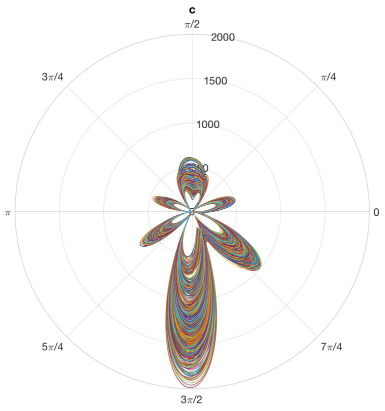


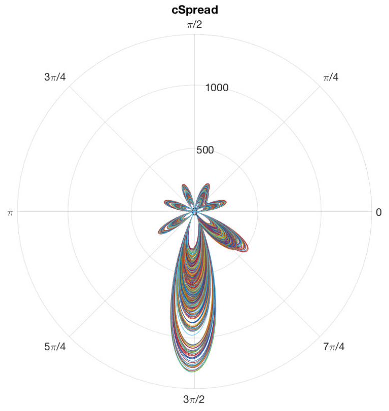
oPursed



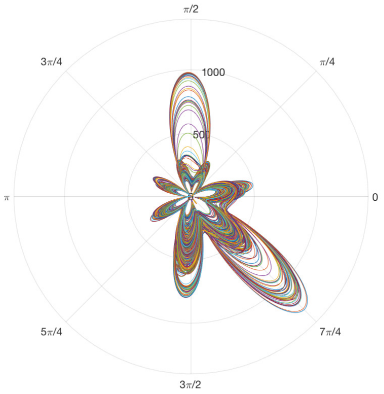

babyO
π/2
600
400
200
3π/4
π/4
π
0
5π/4
7π/4
3π/2


babyOFingers
π/2
1500
1000
500
3π/4
π/4
π
0
5π/4
7π/4
3π/2

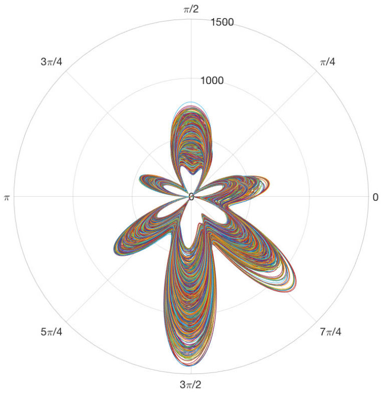

c
π/2
2000
1500
1000
500
3π/4
π/4
π
0
5π/4
7π/4
3π/2


cSpread
π/2
1000
500
3π/4
π/4
π
0
5π/4
7π/4
3π/2

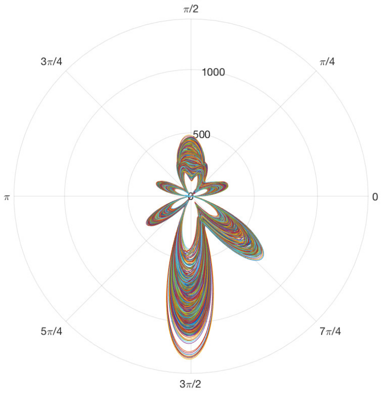
index



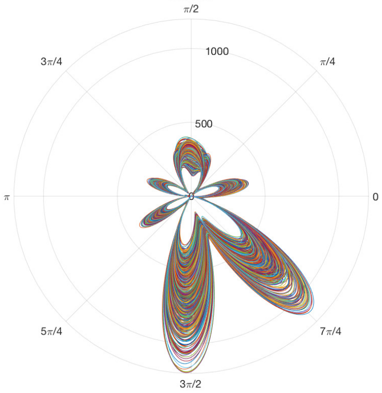
indexL



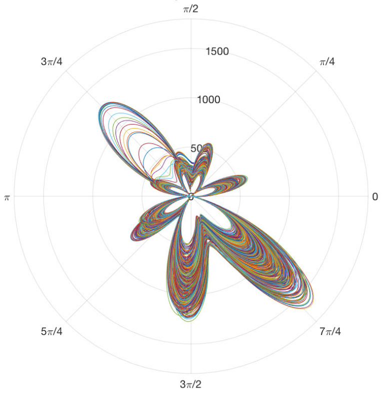
flat



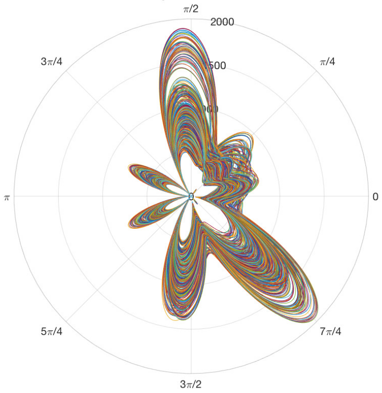
flatBent



spread5

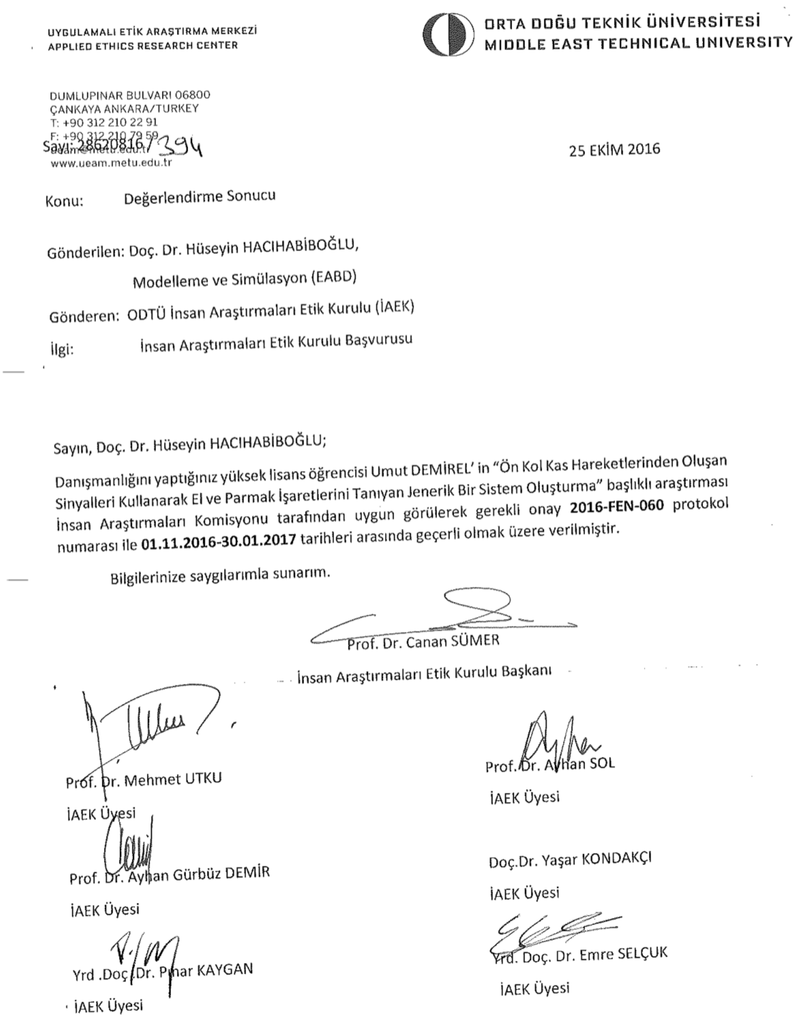


spread5Bent



Appendix B

APPROVAL OF METU HUMAN SUBJECTS ETHICS COMMITTEE



UYGULAMALI ETİK ARAŞTIRMA MERKEZİ
APPLIED ETHICS RESEARCH CENTER

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 22 91
F: +90 312 210 79 59
ueam@metu.edu.tr
www.ueam.metu.edu.tr

Sayı: 28620816 / 394

25 EKİM 2016

Konu: Değerlendirme Sonucu

Gönderilen: Doç. Dr. Hüseyin HACIHABİBOĞLU,
Modelleme ve Simülasyon (EABD)

Gönderen: ODTÜ İnsan Araştırmaları Etik Kurulu (İAEK)

İlgi: İnsan Araştırmaları Etik Kurulu Başvurusu

Sayın, Doç. Dr. Hüseyin HACIHABİBOĞLU;

Danışmanlığını yaptığınız yüksek lisans öğrencisi Umut DEMİREL' in "Ön Kol Kas Hareketlerinden Oluşan Sinyalleri Kullanarak El ve Parmak İşaretlerini Tanıyan Jenerik Bir Sistem Oluşturma" başlıklı araştırması İnsan Araştırmaları Komisyonu tarafından uygun görülerek gerekli onay **2016-FEN-060** protokol numarası ile **01.11.2016-30.01.2017** tarihleri arasında geçerli olmak üzere verilmiştir.

Bilgilerinize saygılarımla sunarım.

Prof. Dr. Canan SÜMER
İnsan Araştırmaları Etik Kurulu Başkanı

Prof. Dr. Mehmet UTKU
İAEK Üyesi

Prof. Dr. Ayhan SOL
İAEK Üyesi

Prof. Dr. Ayhan Gürbüz DEMİR
İAEK Üyesi

Doç.Dr. Yaşar KONDAKÇI
İAEK Üyesi

Yrd .Doç. Dr. Pınar KAYGAN
İAEK Üyesi

Yrd. Doç. Dr. Emre SELÇUK
İAEK Üyesi